(2)

RL-TR-93-5
In-House Report
January 1993

ROME LABORATORY

NEURAL NETWORKS, RELIABILITY AND DATA ANALYSIS

Paul S. Yaworsky, James M. Vaccaro

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-93-5 has been reviewed and is approved for publication.

APPROVED:

ANTHONY J. FEDUCCIA
Systems Reliability Division
Electromagnetics & Reliability Directorate

FOR THE COMMANDER:

JOHN J. BART
Chief Scientist
Electromagnetics & Reliability Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL (ERSD) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1993	In-House Jul 90 - Dec 91

4. TITLE AND SUBTITLE
NEURAL NETWORKS, RELIABILITY AND DATA ANALYSIS

5. FUNDING NUMBERS
PE - 62702F
PR - 2338
TA - 02
WU - 5D

6. AUTHOR(S)
Paul S. Yaworsky, James M. Vaccaro

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rome Laboratory (ERSD)
525 Brooks Road
Griffiss AFB NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (ERSD)
525 Brooks Road
Griffiss AFB NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-93-5

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Paul S. Yaworsky (ERSD)/315 330-4205

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Neural network technology has been surveyed with the intent of determining the feasibility and impact neural networks may have in the area of automated reliability tools. Data analysis capabilities of neural networks appear to be very applicable to reliability science due to similar mathematical foundations. Research on data issues is described, with the goal of providing insight into automating intelligent information processing and data analysis. The development of the Statistical Neural Network is also described, emphasizing how statistics can be used to indicate and help determine natural tendencies in data.

14. SUBJECT TERMS Neural Networks, Reliability, Data Analysis, Automated Reliability Tools, Automated Intelligent Information Processing, Statistical Neural Network

15. NUMBER OF PAGES 68

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Table of Contents:

0.0 Executive Summary 1
1.0 Introduction 2
  1.1 Role Of Automated Tools & Techniques in R&M 3
  1.2 Role Of Technology Development in R&M 4
2.0 Neural Networks Background 5
  2.1 Interest in Artificial Intelligence 5
  2.2 Renewed Interest in Neural Networks 6
3.0 Neural Networks Overview 7
  3.1 Architecture and Operation 7
  3.2 Significant Neural Network Features 8
    3.2.1 Automated Learning 8
      3.2.1.1 Supervised Learning 9
      3.2.1.2 Unsupervised Learning 9
    3.2.2 Generalization 10
    3.2.3 Parallelism 11
  3.3 Technical Challenges in Neural Networks 12
4.0 Neural Networks and Reliability Theory 14
  4.1 Problems With Current Automated R&M Methods 15
  4.2 Neural Network Applications to R&M 16
  4.3 Mutual Areas of Interest - Mathematical Analysis 18
  4.4 Application Considerations 19
  4.5 Potential Benefits 21
5.0 Basic Research in Data Analysis 22
  5.1 Physics, Frequency, and the Nature of Data 22
  5.2 The Data Spectrum 23
  5.3 The Octave Rule 25
  5.4 Harmony of Data 27
  5.5 Future Work 28
6.0 Perspective on What the Research Means 30
  6.1 Automating Information Processing 30
  6.2 The Learning Process 32
  6.3 Harmony, Understanding, Thoughts, and Language 33
  6.4 Reliable Communication - Getting the Point Across 35
  6.5 Data Analysis - Making Sense of It All 37
7.0 Statistical Neural Network - A Prototype Application 40
  7.1 Statistical Network Design 40
  7.2 Network Operation 55
  7.3 Statistical Neural Network in Action 56
  7.4 Summary 60
8.0 Conclusion 61
9.0 Final Remarks 63
10.0 References 65

List of Figures:

Figure 3-1. Typical Processing Element 7
Figure 4-1. Generic Neural Network Model of a Process 20
Figure 5-1. Initial Data Spectrum 24
Figure 7-1. Data Set Generated for Parameter Data1 41
Figure 7-2. Data Set Generated for Parameter Data2 41
Figure 7-3. Data Set Generated for Parameter Data3 41
Figure 7-4. Data Set Generated for Parameter Data4 41
Figure 7-5. Linear Regression Line (1 to 2) 46
Figure 7-6. Linear Regression Line (1 to 3) 46
Figure 7-7. Linear Regression Line (1 to 4) 46
Figure 7-8. Linear Regression Line (2 to 3) 46
Figure 7-9. Linear Regression Line (2 to 4) 47
Figure 7-10. Linear Regression Line (3 to 4) 47
Figure 7-11. Bar Graph of Normal Coefficients (1 to all) 48
Figure 7-12. Bar Graph of Normal Coefficients (2 to all) 49
Figure 7-13. Bar Graph of Normal Coefficients (3 to all) 50
Figure 7-14. Bar Graph of Normal Coefficients (4 to all) 51
Figure 7-15. Bar Graph of Correlation Coefficients Showing Linear Relationships..... (1 to 2), (1 to 3), (1 to 4) 52
Figure 7-16. Bar Graph of Correlation Coefficients Showing Linear Relationships..... (2 to 3), (2 to 4), (3 to 4) 53
Figure 7-17. Statistical Neural Network Architecture 54

0.0 *Executive Summary*

This report documents the results of an in-house effort to determine the feasibility of using neural network techniques in the development of automated Reliability/Maintainability/Testability (R/M/T) tools. While many automated R/M/T tools already exist, and while some may even use artificial intelligence techniques, this effort specifically investigated the field of neural networks as a potential source of automated reliability analysis techniques. Neural networks provide some very interesting and powerful data analysis capabilities, and the technology has become a significant research area in the past five years. However, neural network research and development has had little impact in the field of reliability, with the two areas progressing independently of each other. Most researchers have been interested in performing data analysis for specific applications in their respective fields. The concerns of R/M/T have yet to be addressed using neural networks.

The results of this initial effort indicate that it would be very worthwhile to develop neural network techniques with the goal of improving the overall effectiveness of reliability analysis. Fundamental math-based similarities exist between neural networks and reliability in the areas of probability, statistics and data analysis, indicating that a combination of neural networks and reliability would not only be natural, but useful and powerful as well. This report will introduce neural network technology, characterize significant features of neural networks, discuss problems with existing automated Reliability/Maintainability (R&M) methods, recommend areas where reliability can benefit from the application of neural networks, discuss basic research done on data and related data analysis techniques, present a perspective on what the research means, and describe the design of a neural network whose architecture is based on statistical features of data.

The main purpose of a neural network is to process data such that the network can learn information embedded in data, and once learned, to recall that information in a useful fashion. The most significant results of this work have been a comprehensive understanding of the state-of-the-art in neural networks, and also the realization that underlying components of data exist and can be used to improve many kinds of data analysis techniques. This work has formed a foundation which will aid in the development of more efficient automated R&M analysis tools. While much more research and development is needed, the resulting data analysis techniques will no doubt be useful for very many R&M applications. Many of the advantages of neural networks are especially applicable to reliability theory due to similar mathematical foundations.

1.0 Introduction

The purpose of this effort was to determine the feasibility of using neural network techniques in the development of automated R/M/T tools. The approach taken was to examine the potential benefits of neural network technology at a high level, to focus on underlying principles of neural networks and reliability theory, and then to evaluate the applicability of neural network principles to the R&M problem domain. This investigation led to basic concerns involving the nature of data, and related data analysis issues needed to be addressed before attempting to automate the kinds of issues which surfaced. The reasons for undertaking this task in the first place were apparent similarities in the underlying mathematical nature of neural networks and reliability. Both fields involve data analysis which uses similar mathematical operations and concepts. The math modeling used in both fields are based on probability theory and statistical mechanics, making use of data descriptions and distributions, random processes and uncertainty principles. Besides mathematics, related concepts also exist in physics, engineering, and artificial intelligence. Neural network research itself includes work from the technical disciplines of electrical engineering, computer science, mathematics, biology, neurology, physiology, psychology, physics and others. Neural networks, probably more than any other technical discipline, have actually encouraged and benefitted from many disciplines working together. Reliability has much to gain by being included in this work.

This report discusses neural network features and their characteristics which have application to the broad area of reliability analysis. The goal has been to determine if neural network techniques can be used to perform some of the functions required by a reliability engineer which are currently performed using other (e.g. manual) methods. The initial approach emphasizes neural network techniques implemented in software. This effort does not address neural network hardware or the reliability of such hardware, nor does it address the reliability of software used to perform neural network techniques. The overall goal of this work is to develop useful automated R&M tools and capabilities which currently do not exist.

Section 2 of this report provides a little background on neural networks, indicating how the technology got started. An overview or synopsis of the technology itself is given in section 3. Areas of mutual concern between reliability and neural networks are addressed in section 4. Section 5 describes research done in this effort concerning data and related data analysis issues. A few more words should be said about this research. While not envisioned initially, research issues surfaced which addressed basic needs lacking in the area of automated intelligent information processing. Underlying components of data appear to exist and be exploited in the brains of animals naturally. With one of the goals of neural networks being to automate functions similar to those of (animal) brains quickly and efficiently, we have tried to characterize some of the underlying components of data. Our emphasis was on frequency aspects of data. Attempts initiated here have had interesting results and implications. The impact that this work may have on automating information processing systems is of course unknown, but the potential is enormous. More research and development is needed to explore possible directions and applications. The work is described in an introductory nature in sections 5 and 6 of this report.

Another part of this work has been the development of a neural network which relies on the statistical nature of data to build its architecture. The Statistical Neural Network, as it is called, uses data descriptors to help design the layers, nodes, and connections of the network's architecture. The Statistical Neural Network is described in section 7, with an example provided to help explain the operation of the network. Section 7 addresses one of the fundamental links between neural networks and reliability, namely statistics.

1.1 Role Of Automated Tools & Techniques in R&M

In the past, reliability engineers have specialized in developing and applying reliability and maintainability principles in order to satisfy the reliability requirements for the products they've worked on. Over the years, many kinds of reliability methods and techniques have been used. These tasks have relied heavily on sound mathematical principles, good data, and a manual process to make sense of it all. But as computers have become more and more widespread, they have come to be used by reliability engineers to do the number crunching and other types of data processing tasks which are so much a part of their work. This has caused a shift away from the manual, task-oriented nature of the work to a more automated, process-driven way of doing things [18]. To be sure, the same kinds of reliability tasks will still have to be performed, as the requirements for them have never been more necessary. Today's Air Force avionics have very high reliability requirements. Over time, avionic systems have become much more complex. The combination of many complex components and the difficulty of analyzing how they interact with each other have led more to automated analysis. While the need for reliability work will not go away, the nature of that work is slowly changing. Reliability engineers are taking advantage of computer hardware and software, among other things, to ease the burden of the math and data intensive analyses required of them. Thus the overall impact of automated data processing is having a positive effect in the R&M community with the aim of performing the necessary tasks in a quicker, easier, more accurate fashion.

1.2 Role Of Technology Development in R&M

Computer technology has come to play a major role in the engineering community. With the goal of making the job of the reliability engineer more practical, efficient and accurate, we pursue the development of automated technologies. There can be no doubt that computers have proven beneficial to the field of reliability. Technology development for automated tools is needed in order for reliability science to maintain state of the art. Automated technologies such as Computer-Aided Design, Computer-Aided Manufacturing, and Finite Element Analysis have provided capabilities that would be impossible to perform manually. Research and development (R&D) is necessary to enable the development of advanced tools and technologies. Rome Laboratory has been involved in R&D for many kinds of automated R&M tools [9].

Most of the computer hardware used for R&M is commercially available. Very few, if any, features or capabilities of conventional computers are specific to the field of R&M. Much of the computer-related research involves the generic capabilities of computers, such as using commercial state-of-the-art hardware or software, integrating existing techniques in a novel way, or developing better procedures or algorithms which run on general purpose machines. However, certain aspects of neural networks appear to have specific significance to R&M. The mathematical similarities of neural networks and R&M, right down to their fundamentals, imply that with proper development, neural networks can provide enormous benefits to the field of reliability. By examining where neural networks have come from, seeing where the technology is today, and envisioning future capabilities, this report will characterize neural networks as applicable to the field of reliability.

2.0 *Neural Networks Background*

Neural networks are a subfield of artificial intelligence (AI). The other subfield of AI can be called traditional AI. Both disciplines involve automating functions which, if performed by a human, would require intelligence. Thus all of artificial intelligence tries to mimic certain aspects of human intelligence using computational methods. Traditional AI believes that intelligence can be programmed explicitly using symbolic languages and formal logic. It is a top-down approach which places much emphasis on computer science and programming. Neural networks have more of a neurobiological origin, with emphasis on trying to model living neural functions. The two main interests of neural network research are in developing realistic biological models and in developing machines (computers) which perform intelligent functions. The latter is our concern, with the belief that a network of relatively simple processing elements connected in some complex yet orderly fashion can be used to represent and perform intelligent-like functions. Neural networks represent a bottom-up approach to computation, making use of simple processing elements connected in a complex parallel fashion. The approaches of traditional AI and neural networks are quite different. Yet both disciplines work toward overlapping, if not similar, goals. The task of automating intelligent functions is enormously complex. The factors involved are many, and the functions being automated are not very well understood. Toward this extraordinary goal, researchers have only scratched the surface.

Artificial intelligence began as a formal discipline in the summer of 1956 with the Dartmouth Summer Research Project on Artificial Intelligence [20]. Also at this conference, the field of neural computing was launched [17]. Today researchers interested in neural networks may have a background in mathematics, biology, psychology, neurology, cybernetics, control theory, engineering, information theory, physics, cognitive science, computer science or various other related disciplines.

2.1 Interest in Artificial Intelligence

Although traditional AI and neural networks have been in existence for over thirty-five years, the amount of work done under the two disciplines has varied considerably. Traditional AI has enjoyed steady interest over the years, yet its approach and especially its progress have been controversial. One of the more successful areas of traditional AI are expert systems. These systems have been accepted in many applications where expert knowledge can be well defined and explicitly encoded as sets of decision rules. As automated technologies have evolved, expert systems have matured to the point where it is questioned whether they still belong under AI. In any case, expert systems are still of interest to traditional AI researchers, along with areas such as knowledge-based systems, automated planning, programming and reasoning, natural language processing, validation and verification of software, and intelligent computer interfaces.

Neural networks have not enjoyed steady interest since their beginning. Lack of knowledge of neural-like functions, insufficient math models, and the state of hardware and software technology severely limited progress. Limitations of early neural models, and particularly of Frank Rosenblatt's *perceptron*, were documented in 1969 by Minsky and Papert in their influential but controversial book [21]. Minsky and Papert's book, while mathematically thorough, drew some harsh conclusions about *perceptrons* Among other things, they implied that neural networks more complex than those analyzed in their book were of little scientific interest. Largely as a result of this well-written but misleading book, interest in neural network research, and the money which funded it, dropped [26].

2.2 Renewed Interest in Neural Networks

For the next fifteen years or so, relatively few researchers worked in the area of neural networks. But by the mid-1980's, several developments had combined to renew interest in neural networks. Better understanding of some of the brain's functions led to better neural network models. Newer models used more appropriate math functions and techniques such as nonlinear transfer functions. Multilayer architectures overcame the limitations of single-layer perceptrons. Better forms of knowledge representation were being developed, and advanced computer hardware and software technologies were providing the computer power needed to perform complex neural network simulations to a wide variety of researchers. The results of research drew more and more interest, and soon neural network technology grew so big so fast that today there is a flood of interest and material on the subject. Current indications are very promising that as neural network technology develops, the resulting data analysis capabilities will be useful for very many applications. However, while the promise stands, much more work is needed before neural networks can enjoy widespread acceptance.

3.0 Neural Networks Overview

A neural network is a math-based, neurologically inspired model used to perform certain kinds of data analysis. The architecture of a neural network, as well as methods of operation, will be described next. This will be followed by a discussion of the features which make neural networks so interesting. These features are directly related to network architecture and operation.

3.1 Architecture and Operation

The architecture of a neural network generally consists of layers of processing elements. The processing element is the basic building block of the network. A typical processing element is shown in figure 3-1. Mathematical functions are used to represent a transfer function which maps the element's parallel input signals to an output signal. Many signals come in, get combined, and then pass through a threshold function which determines the output value. The output is connected to the input of many other processing elements, creating a layered network. Each connection in the network is represented using a mathematical quantity which allows a *weight* to be associated with that connection. These *weights* are used to represent information, and are a most essential concept in the network learning process. The *weights* are modifiable, allowing network connections to be strengthened, weakened, left unchanged or even eliminated during operation. The network connection scheme and the number of layers, processing elements and inputs per processing element combine to create many possible kinds of architectures.

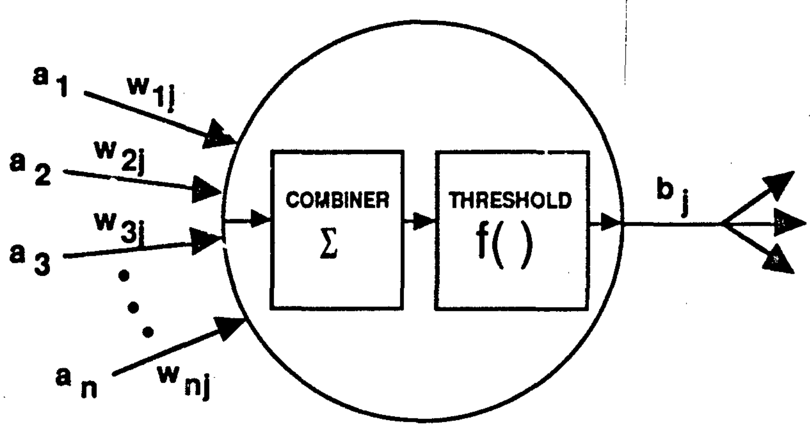


Figure 3-1. Typical processing element. The processing element multiplies each input signal by its appropriate connection weight. Signals get combined and passed through a threshold function, producing one output signal.

The operation of a neural network consists largely of methods to control learning and recall within the network. Learning involves assigning or modifying the connection weights, which is basically inputting and encoding information in the network. Recall is the process of accessing or outputting information, and provides the network's response for a given input. Operation also involves such issues as network initialization, state values, timing, input and output requirements, and methods to monitor and control the functions of the network. More details of neural network operation will be given in the following sections.

3.2 Significant Neural Network Features

The most significant features of neural networks come under the heading of information processing, and more specifically, data analysis. Neural network analysis features will be discussed, with emphasis on how they may complement existing methods. While the following neural network features do exist, researchers are working toward making them more practical. These features have become evident by examining the literature and compiling and correlating the results of many independent researchers. The most significant characteristics of neural networks are that they:

- **learn**
- **generalize**
- **use a parallel, hierarchical architecture**

Each of these will be discussed in more detail.

3.2.1 Automated Learning

Learning is defined here as lasting change toward improved performance resulting from experience. The lasting change is handled by the network's connection weights. Improved performance involves goal-directed response which is controlled by a learning procedure or algorithm. Experience is data. These definitions can easily lead to others, such as for information, knowledge, intelligence and understanding. Suffice it to say that learning involves change in something, and that change ought to have a purpose. The ability to learn is by far the most significant aspect of neural networks. This concept is not only powerful, but it is an essential part of information processing. While conventional computers do many things, having established a firm place in our society, if they learn at all (as in some forms of conventional AI), they do so very awkwardly.

Neural networks learn, or are trained, as data passes through the network. The connection weights get modified according to the designed-in learning procedures. Changing data causes the network to change its weights. Programming is implicit, being a function of the data and the network's architecture. This contrasts with the programming method of conventional computers, which is much more explicit. While explicit programming has its advantages, the real world is not explicitly understood, and consequently very many situations exist which need better solutions. The possibility of neural networks filling this need is very good, since it includes the concept of learning. Neural network learning is often classified as being supervised or unsupervised, each of which will be described next.

3.2.1.1 Supervised Learning

Supervised learning, as the name implies, involves presenting the network with some kind of supervision during learning. Usually this is accomplished by providing the desired response to the network as part of its training data. Weights would then be changed based on the difference between the network's actual and desired responses. The goal of training here is to have this difference converge to a minimum value. Convergence is a very important concern, with the goal of having the network converge to a global rather than a local minimum. This form of learning tends to take a long time and be performed off-line, since satisfying convergence criteria can involve many iterations. However, when properly applied, solutions also tend to be quite good, given that known solutions are used as part of the learning process. Many kinds of supervised learning methods exist, and different versions of each kind may have different names. For example, the difference method described above may go by the name error correction, delta rule, least mean square error rule, etc..

3.2.1.2 Unsupervised Learning

Unsupervised learning involves having the network determine weight adjustment itself, without any supervision. The network does this mathematically by organizing it's weights as a function of how new input data is related to or associated with previously input data. Connection weights may be initially set to random values, and as the network learns, or trains, it will accumulate or distribute its weights accordingly. Similar training patterns will reinforce existing weights, and new patterns will form new connection patterns. Complex relationships can be constructed this way. Hebbian learning, first introduced as a technique for learning in biological neurons [15], has been used as a basis for this kind of learning. Hebbian learning basically states that if an input and the output of a processing element are large, then the weight adjustment for that particular input should be large. Thus the input and the output are correlated such that the corresponding input weight is strengthened (or weakened) accordingly. The processing element becomes more sensitive to similar patterns at that input. This has become an important learning law in neural networks. Variations of Hebbian learning may even be modified to produce supervised forms of learning. In fact, all forms of learning use some kind of criteria to adjust weights, and it becomes a matter of definition as to what supervision actually is.

Networks based on unsupervised learning are often used for classification or as associative memories, since new data can be associated with data already present in the network. Many variations of associative neural networks exist [26]. In fact, all neural networks have some associative nature to them. Dr. James Anderson of Brown University has stated that association is one of the foundations of human cognition [4]. The ability of the networks to learn data associations and to represent complex data relationships is one of their most significant features. Since unsupervised networks organize or modify their weights based solely on input patterns, learning tends to be quicker, and perhaps less accurate, than in supervised learning. Unsupervised learning techniques are used for on-line or real-time applications where weight adjustment is designed for convergence with relatively few iterations.

Another kind of learning is called graded learning. Here the network uses feedback of some sort to determine how it is doing (good or bad), but it is not given the desired response, usually because it is not available. This puts graded learning somewhere between supervised and unsupervised. Learning methods can also be combined in a network to form more complex learning algorithms. Different layers of a network can use different learning methods. A technique called competition can also be designed into a neural network. This allows processing elements to compete for the privilege of learning. The winner of the competition adjusts its weights and output accordingly, and the losers are prohibited from learning. Technical challenges concerning neural network learning will be discussed later in this section.

3.2.2 Generalization

Another significant feature of neural networks is their ability to generalize. This means that the network can produce a general response to an input or set of inputs it has never seen before. This feature has to do with the network's ability to make associations and approximations between its many stored patterns. If input data is noisy, if part of its content is missing, or if it just hasn't been learned yet, the network will generalize with its "best guess". New or novel data is handled by having the network respond with an output which is most closely associated with patterns already stored. This requires neural networks to be able to deal with fuzzy concepts. Fuzzy logic and fuzzy sets, along with other areas of mathematics which allow approximations and averages to be made, help enable the generalization capability.

The ability to generalize is handled quite naturally in neural networks, while it is not handled so well in conventional computers. This is because conventional computers are based on boolean logic, with everything being either "1" or "0", black or white. They are designed to be very precise and logical. The accuracy of conventional computers conflicts with the very concept of generalization. Neural networks, in theory, are designed to handle data as it comes in from the real world. Their quantities may be discrete or continuous. The quality of data may be noisy, fuzzy, or incomplete. A goal for neural networks is to be able to handle real world data in a form most closely to that which it naturally occurs. This requires the ability to generalize.

Part of the reason neural networks can generalize stems from their parallel architecture. By having many parallel inputs, each processing element can integrate many signals at once. Techniques such as averaging, thresholding, normalizing and interpolating aid in the generalization process. Data variations can be used to create ranges, and frequency of occurrence can be used to compile statistics. Data entering the network gets distributed and represented as patterns of connectivity. These patterns come to represent general forms of data, having many links and weights associated with them. This kind of operation is much different from the serial operation typical of conventional computers. The generalization capability of neural networks is by definition a general feature, leading to more specific capabilities such as association, classification, estimation, optimization, and recognition. The downside to the generalization feature can be misleading or incorrect results.

3.2.3 Parallelism

The third significant feature of neural networks, already mentioned above, is their parallel architecture. While parallelism tends to be a very complex feature to design into a system, it will undoubtedly be an essential part of the most advanced systems. The parallel architecture of neural networks makes it possible to represent and process complex relationships quickly and efficiently, adding functionality not present in serial computers. While each processing element may be mathematically simple, the parallel configuration of these simple processing elements can allow complex, powerful behavior to be achieved at a higher level. The hierarchical structure of the network enables a global view of data at the highest level, with complex data patterns broken down via simple processing elements connected in a parallel fashion. The parallel architecture of neural networks should not be underestimated. The parallel architecture comprises the "hardware" of neural networks. The inspiration from living neural systems may eventually provide the insight to enable researchers to overcome the complexities facing parallel computer design.

Another aspect of parallelism is fault tolerance. By distributing data, processing, and interconnect across its entire architecture, no one area of the network is critical for operation. While this is typically true at the middle layers of a neural network, the input and output layers tend to be less fault tolerant. It appears that fault tolerance is directly related to the degree of parallelism involved. Also associated with parallelism in neural networks is the characteristic of graceful degradation. This means that as failures tend to occur in a network, its operation degrades less abruptly or drastically than for serial approaches. Graceful degradation implies that a faulty network can provide an output that is less than optimum but still useful.

The various features of neural networks combine to form very interesting systems. The ability to learn, the ability to generalize and process real world data effectively, and the functionality of a parallel structure consisting of simple processing elements connected in a hierarchical fashion form the basis of neural networks. Neural networks are fundamentally very different from conventional computers. However, the two types of computers are not in competition with each other. In fact, they may very well complement each other in future systems.

3.3 Technical Challenges in Neural Networks

Neural networks are not a mature technology. Many technical challenges remain, some of which can explain why the technology has not achieved widespread acceptance. The biggest challenge by far is neural network design. Very many different designs exist, but each has its shortcomings. No single, general purpose design exists which is powerful and efficient enough at solving a wide variety of problems. Also, given the large number of people working this relatively new problem, lack of consensus exists over what the various neural network terms, concepts and techniques actually mean. This adds to the confusion in this already complex technology. Consequently, neural networks may be incorrectly applied where they might have been useful, or they may be used in areas where they do not apply. Researchers are currently trying to standardize various aspects of neural network technology.

The main challenges of neural network technology stem from the difficulty of obtaining useful and efficient network designs. Designs principally involve network architecture and operation, with learning algorithms at the heart (or actually, the brain). Due to the complexities of building in powerful functions, necessary control, and useful features, the state-of-the-art is relatively immature at this time. However, it is believed that concepts stemming from those being researched today, especially in the areas of learning and control, will one day be used to perform functions similar to those of living neural systems.

The underlying concept of man-made neural networks is that they readily exploit the mathematical properties inherent in data. The networks represent and manipulate *information* embedded in data. To be able to do this, we assume that data inherently contains properties which can adequately be described in mathematical terms. Considering the nature of data, from the physics of its source to the statistics of its content, it appears that this is a good assumption. It would then follow that the mathematical properties inherent in data can readily be exploited. These issues are discussed in more detail in sections 5 through 7 of this report.

Another challenge in neural network design has to do with the availability of good data. Data is needed to train the network, and usually different data is needed to test or run the network. The learning which takes place during training hinges upon the form in which data is represented in the network. Representation may involve binary data (only 0 and 1, which is used in digital computers), analog data (which is the form of most naturally occurring data), or fuzzy data (many values between 0 and 1). An area which has not been developed well enough is the one concerning the mathematical nature of data. It is believed that basic concepts stemming from the laws of physics and described in the language of mathematics can be used to better characterize data. Preliminary research indicates that data representation and processing may be done using techniques based on the frequency components of naturally occurring data. By breaking data into its fundamental components (frequency, phase and amplitude), more efficient techniques may be developed to process and analyze real-time data.

The development of appropriate learning methods and algorithms are by far the most intriguing and elusive aspect of neural network research. Unfortunately, no universally useful or efficient automatic learning method exists. Given the overall complexity of the task, it appears that the field of neural networks will progress slowly. Current neural networks may have processing elements which number in the 1000's or 10,000's. The average human brain has over 10,000,000,000 neurons [27]. Each living neuron is much more complex than its electronic counterpart. Also, as the number of processing elements increases, the complexities of connecting them together in a meaningful way becomes enormous. The physical problems encountered when implementing these interconnects in hardware approach the impossible, given today's technology. If it were not for living proof that these kinds of systems exist, interest in building man-made neural networks might have faded by now. It is the extraordinary undertaking of trying to emulate the brain that provides motivation in building intelligent machines. The resulting data analysis techniques will no doubt be useful for very many applications.

4.0 *Neural Networks and Reliability Theory*

While conventional computers have been used in the development of analytical R&M tools, neural networks have not yet been applied to the R&M problem domain. In fact, very little work has been done investigating the potential benefits neural networks may offer to the field of reliability. Neural networks will of course have unique reliability concerns, as all technologies do. The reliability of neural network hardware will have to be addressed. Also, fault tolerance is a feature of neural networks often mentioned, but relatively little work has been done to characterize or maximize the fault tolerance of the networks [8]. This effort, however, is not concerned with the reliability of neural network hardware or software. It is concerned with the specific task of using neural network technology in the development of better analytical tools for reliability assessment. Many kinds of analysis tools, methods and techniques exist, but the unique features of neural networks may offer additional capabilities which can improve reliability modeling, prediction, measurement and analysis. The techniques discussed here are most often implemented in software and run on existing conventional computers. Various issues concerning neural networks and reliability will be discussed in this section.

The data analysis performed by neural networks is very statistical in nature. The statistics of data get compiled, associated and represented inherently by the network. Research in the U.S. stresses this aspect of neural networks. Tom Schwartz calls neural networks a statistically based mapping technology [25]. Rumelhart, McClelland, and the PDP Research Group describe neural networks as simple, parallel processing elements which perform complex statistical processes [24]. European researchers, on the other hand, stress probability theory and arithmetical logic more in their implementation of neural networks [1]. In general, all neural networks compile statistics and form data distributions based on training data. The literature is full of examples of neural networks which represent and process many kinds of data and perform a wide variety of analysis functions. What has not been emphasized well enough is that underlying neural network technology are statistical and probabilistic techniques which form the basis of their operation. Many other areas of mathematics are also used in neural networks. Since the networks are math-based, virtually any area of mathematics could be incorporated into a neural network. In any event, statistics and probability provide the foundation for the data representations and relationships which neural networks ultimately model. Reliability theory is also based heavily on probability and statistics. Thus this effort addresses the overlap between neural networks and reliability theory.

4.1 Problems With Current Automated R&M Methods

Many current R&M methods include a fair amount of automation to perform their math-intensive analyses. The majority of these automated methods are run on conventional computers, and are often quite useful. However, conventional techniques are limited by the restrictions of the machines they run on. These restrictions include:

- programming methods do not adjust to variations in data
- digital precision does not easily allow generalization
- serial operation limits complexity of data processing abilities

Other limitations of current automated methods are due to the uncertain nature of data in the field of reliability:

- difficulties in assigning probabilities for reliability models
- excessive or incorrect use of assumptions

Also, the state of the art in computer science and artificial intelligence has difficulty extracting *information* from data. Too little insight is provided into what data actually represents. This is where the manual process of data interpretation and analysis has been used. An additional restriction to current R&M methods in general is the emphasis on testing and after-the-fact assessment of parameters important to quality rather than on identifying and eliminating the root causes of defects early-on.

While all these problems may not necessarily be solved by neural network technology, especially given its state-of-the-art, neural networks can indeed address them and perhaps lessen some of the current restrictions. Neural networks can be thought of as a tool for modeling different kinds of data analysis problems. How this tool develops, and ultimately how it gets used, remain to be seen. It appears obvious that no one computer method will solve all problems, and that combinations and interaction of each useful method is a viable approach. This effort stresses that neural networks are at least a part of this approach. Techniques which use conventional philosophies and styles of programming can and should be combined with neural network techniques in future research and development.

4.2 Neural Network Applications to R&M

This section discusses how neural networks can be applied to the field of reliability by listing each of the problems mentioned in section 4.1 and addressing it with the appropriate neural network feature(s).

Programming methods do not adjust to variations in data. Current statistical methods do not adapt well to changing data. The methods may not be flexible enough, or they may require manual interaction, to handle dynamic or unexpected data values. Neural networks rely on changing data to adjust weights according to the inherent statistical relationships of input data. This area of application stems from the fact that neural networks are not programmed explicitly, but learn data associations which are implicit in the input data patterns.

Digital precision does not easily allow generalization. Digital computers are great at performing precise calculations and formal logic. However, they do not handle noisy data very well. Missing data often causes havoc. In general, conventional computers are rigid, precise machines which must be programmed exactly. This is required for a great number of applications, including many in reliability. But debatably, this aspect of conventional computers greatly limits their use. While the general, less formal, uncertain methods of computation and data processing used in neural networks are not well developed, they offer alternative solutions and possibilities of addressing many applications not yet explored.

Serial operation limits complexity of data processing abilities. Conventional computers operate serially at extremely high clock rates. This works well for very many applications, including neural network implementations which are run on conventional computers. However, neural networks distribute processing over many elements instead of through one central processing unit. A hierarchical structure of relatively simple processing elements connected in a layered, parallel fashion offers functionality not present in conventional computers. An additional aspect of parallelism not yet fully realized in neural networks is that they allow parallel inputs to be combined simultaneously at each processing element. This creates multiple levels of parallelism within a network, which brings potentially much computer power.

Difficulties in assigning probabilities for reliability models. In general the level of accuracy or confidence associated with reliability depends on the values of basic parameters used in its determination. Reliability theory accounts for probabilities and confidence limits, but does not say how to assign these values in the first place. Neural networks can provide data analysis techniques which characterize and process basic parameters which reliability models require. A natural way to handle probabilities using neural networks would be to let each connection weight, as it forms, represent a probability. Network learning, which consists of adjusting weights, could be made to automatically reflect probability distributions of input data.

Excessive or incorrect use of assumptions. Assumptions are often useful in estimation and prediction, but bad assumptions lead to incorrect or misleading results. While humans can rely on experience to make skilled assumptions, conventional computer methods typically try to encode massive amounts of explicit data, or program rule-based systems which lead to solutions for very limited domains. Given the same available data, neural networks tend to make better generalizations and approximations. The weighting schemes used in neural networks handle data by scaling the relative importance of data, based on the network's learning algorithm. Assumptions will still have to be drawn, but the network would be extracting or utilizing more of the information embedded in the data, making the human operator's job easier.

Difficulty extracting information from data. Too little insight is provided by conventional methods into what data actually represents. Inferential statistics, together with neural networks, can provide insight into data patterns and their embedded information. These techniques may not suffice by themselves, but they can certainly aid in the manual analysis process, and in some cases even improve on it. This stems from the fact that neural networks try to make use of all available data, not just prepared data. Neural networks are often used for feature extraction or as filters which preprocess data for more conventional computer techniques.

General emphasis on testing and after-the-fact assessment. The philosophy in reliability has traditionally been one to determine or predict how reliable a product is or will be by emphasizing *effects* more than *root causes*. This often involves characterizing or testing for problems or failure mechanisms which already exist. The emphasis has been on failures more than on the underlying defects which cause them. A different philosophy is that which is common to such methods as Design Of Experiments, Building-In Reliability and Statistical Process Control. These methods address reliability very early in the life cycle, and try to understand and eliminate the causes of problems, thus preventing them from occurring at all. Neural networks can be used in the data collection and analysis needed to accomplish this. By defining a process, controlling it and improving on it, the statistical methods built into neural networks can be used to model the process and to minimize variance, helping to efficiently produce quality products by design. Neural networks lend themselves to the dynamics of change needed for continuous improvement.

Certainly all the neural network features mentioned above are not specific to reliability applications, but they are directly related. Neural network techniques will be subject to technology limitations, but hopefully some of these limitations will be overcome as the technology matures. In any case, by combining the advantages of neural network technology to those of conventional computer technology, many of the individual technology limits can be overcome. No doubt many R&M application areas exist today which could use the combined strengths of neural networks and conventional computers.

4.3 Mutual Areas of Interest - Mathematical Analysis

Most, if not all, of the mathematical methods used in neural networks today have existed for some time. Applicable areas of math include algebra, geometry, calculus, differential equations, communication theory, control theory, information theory, automata theory, arithmetical logic, formal logic, fuzzy logic, statistics, probability, randomness, uncertainty, and chaos theory. What neural networks contribute is a mechanism which allows many mathematical and other concepts to be combined into one model in a useful fashion. The functionality is such that it brings the power of mathematics, the versatility of data analysis, the architecture of neural systems, the concept of control theory and the ability to learn, among other things, together in one model.

The main areas of interest mutual to reliability and neural networks are probability and statistics. Distributions and data descriptions form an interesting and powerful starting point in the data analysis domain. Probability and statistics may be used to describe data relationships and to help characterize the quantitative and qualitative nature of data. Reliability has been defined as the probability that a system will perform its intended function under specified conditions for a certain length of time or for a certain number of cycles [32]. This effort has indicated that the fundamental concepts of reliability and neural networks can be developed in such a way as to use the functionality of neural networks to perform the analysis and information processing needed to determine reliability.

Probabilities are associated with variables when their values are not explicitly known, either because they could not be measured accurately or calculated precisely enough. Reliability problems are ultimately caused by failures. Failures do not occur randomly, but are caused by defects. Defects are characterized using random processes, along with the many other tools of reliability. Reliability not only involves methods to characterize failures but also those to determine the conditions in which a system should not fail. With uncertainty and indeterminism involved, probabilities, random numbers and statistical methods must be used in the characterization process. Neural networks, more naturally than conventional computers, can be used to represent these kinds of data characteristics. The weight vectors which comprise neural network connections can be used to model probability distributions of features found in input data. Resulting distributions can be used to represent and help determine many parameters such as reliability, mission time, and mean life. Neural networks can adapt their weights according to statistical associations of input data. Techniques which involve averaging, approximations, assumptions and confidence intervals can be used to describe and evaluate data, aiding in the analysis and decision making process. Statistical tools could be used to characterize, control, and improve a process, thereby reducing its variability. Neural networks can be used to extract information from data, allowing inferential statistics to be automated in a more efficient fashion.

Reliability is not an exact science. Its data and related analyses are subject to much interpretation. Neural network features seem to lend the technology to much of the mathematical analysis needed for reliability. Neural networks work with indeterminism, can be used to make estimates or best approximations, and can adjust weighted parameters due to changing data. Potentially many neural network applications exist today in the field of reliability, but the technology will take time to mature. Neural network designs which satisfy fundamental concepts in math, physics, and other basic sciences will be very useful and can be developed to apply directly to the field of reliability.

4.4 Application Considerations

A general understanding of neural networks is necessary when considering the technology for possible applications. Depending on the sophistication of the application, one can use "ready made" neural network hardware or software solutions, develop an application using a neural network development system, or create a network from scratch. The complexity of the task increases quickly in the order given above. Many considerations enter into the picture no matter what the level of complexity or application. A difficult aspect to deal with is the changing nature of the technology. Not only is it relatively immature, the technology involves adaptive techniques which tend to be difficult to design or control. This makes it especially difficult to commit an application to a hardware solution. While neural networks are finding their way into more and more commercial applications, widespread acceptance and usage hinges upon the development of more practical neural network designs.

One big consideration in applications is the *process* actually being modeled by the neural network. The functionality of a neural network can be described at the highest level as shown in figure 4-1.

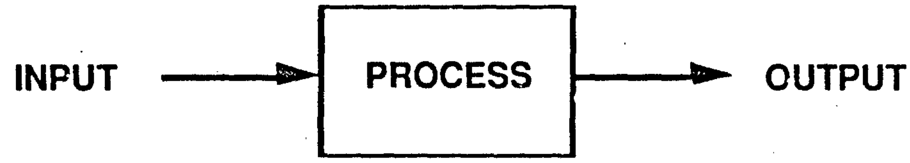


Figure 4-1. Generic neural network model. The network models a process by representing the output as a function of the input(s).

The *process* can be one of many things, such as a mathematical function or equation, or it can be virtually anything that comes under the general definition of the word "process". The process gets embodied in the neural network architecture, and the operation of the network is such that it models or represents the output of the process as some function of the input. Most often, processes contain many inputs and not as many outputs. Math is ultimately the language used to describe the process. The neural network does all the bookkeeping and manipulations necessary to form data associations. A key feature of the model shown in figure 4-1 unique to neural networks is that the process is adaptive. This, combined with the statistical nature of neural networks, makes the technology particularly applicable to Statistical Process Control.

Since neural networks are a data analysis technique, many applications are involved with data collection and processing of raw data. Other applications may take conditioned data and process it for a specific purpose or function. Once a process has been identified as a potential neural network application, several considerations must be examined before the details of network design, monitoring and control can be addressed. First of all, it should be determined whether or not a neural network technique is applicable to the problem at hand. The nature of the problem, the existence of other solutions, the time-frame for development, and the level of complexity involved all affect the decision. When it appears that neural networks offer enough technical advantages to pursue development, the following should be taken into account before choosing a particular neural network:

- define the process to be modeled
- determine the number of inputs, outputs and related parameters
- consider the type of data available, including source, quality, confidence levels, acceptable values, limits, initial conditions
- consider stability and convergence criteria for the process
- will the process be run on or off line? and does it involve time-sensitive data?

After considering the nature of the task at hand and the details of the process, a network has to be chosen which will provide the functionality needed. Very many types of neural networks exist, with others being developed continuously. Some popular neural networks have many variations or options. Other applications involve the design of custom networks which emphasize particular properties and features o. interest. The choice of a particular neural network will consequently define the network's architecture and method of operation. Next, one has to determine which path of in ., mentation is most appiicable. Some neural networks are available in hardware form, but most are . ailable in software (source or executable code available as a specific commercial application, a development system package, custom design, etc.). The method of implementation often depends on how much time, money and manpower is available for the task, as well as the complexity of the application. Finally, the user/designer of the neural network has to be able to train the network and see how it performs.

Another concern for reliability applications is where in the life cycle process does the application lie. The application should focus on either the requirement stage, specification stage, design, test, production, operation or support stage, etc. Each stage has specific parameters, concerns, and characteristics which must be modeled. In reliability, many of the models used in different life cycle stages are related, since parameters such as failure rate or Mean Time Between Failure may be used in different parts of the life cycle. Other more generic applications of neural networks which definitely apply to the field of reliability are to: filter out noisy data, help determine the significance of data, identify data out of range, fill in for missing data, establish defaults, improve on worst case values, and solve number intensive problems which are currently done using graphical or other mathematical methods.

4.5 Potential Benefits

Neural networks bring with them an array of interdisciplinary methods which can be used to represent and solve many kinds of data analysis problems. In particular, the problems dealt with in reliability are especially applicable to the types of analyses performed well by neural networks. With probability and statistics as common threads at the fundamental level, reliability and neural networks form a natural pair. The benefits of using neural networks to perform reliability analysis functions are increased automated capabilities, improved analytical efficiency, increased accuracy, and adaptability. Each one of these alone would be a worthwhile achievement. The combination of them carries enormous potential. The most significant feature of neural networks is their adaptable nature. The ability to learn or adjust is the most powerful, desirable feature one could build into a system. Neural network technology is in its infancy, but it still remains at the forefront of endeavors to automate learning in machines. The long term goal of this work is to develop automated tools and techniques which analyze R/M/T data more effectively. Neural networks can provide the insight as well as the mechanism to achieve this goal.

5.0 Basic Research in Data Analysis

The research done in this effort initially concentrated on data - plain and simple, raw data. From its source to its destination, we have tried to characterize the nature of data by emphasizing its frequency components. Using computer tools, statistics and other forms of mathematical analysis, our multidisciplined approach has examined existing data analysis techniques and has taken a hard look at the concept of data itself. What has come out of this fundamental approach has been a better understanding of the big picture of information processing, as well as insight into some of the details of data processing. Some very interesting concepts and theories have come about. The main results appear in this section, with a perspective on what the research means in section 6. The topics in this section include a brief discussion on the nature of data, how resulting ideas led to a description of the nature of data in the form of a frequency-based spectrum, an octave rule which is used to help focus and filter data, and considerations on the harmony of data and its importance in the recognition and interpretation of data.

The concept of data makes no sense or means nothing if not put in the context of an interpreting network or system. Here the system is a machine (computer), and the goal is to build intelligence into the machine. As part of a thorough investigation of data analysis, physical sources of data would have to be considered. In this effort the sources of data have included music and images (colors, shapes, sizes, etc). Concentration on frequency components of data has indicated that a correlation exists between information processing systems and the underlying physics of the data signals involved. The idea of an orderly structure and nature to data, in its various forms, began to take shape. Mechanisms to focus, interpret and communicate these forms of data were proposed. Examination of human cognition and how it seemed to handle data communication, learning, and understanding provided valuable if not novel insight. With beginnings in automated technique going from artificial intelligence to neural networks, and with sights set on reliability, going from reliability theory to analysis techniques, we have examined the nature of existing techniques, dealt with abstract research topics, and dreamed of a science which involves much more capable computing machines. With a humble start, we now introduce the main topics of our research, the details of which are the subject of future work.

5.1 Physics, Frequency, and the Nature of Data

The physical sources of data examined have been audio signals in the form of music, and visual signals in the form of images, focusing on colors, shapes and sizes. In considering the various forms of data, underlying components and relationships appeared to exist, leading to a particular interest in frequency aspects of data. Further investigation continued to fuel ideas that data inherently contained properties or components which must be exploited in order for intelligent data processing to occur. In order for our brains (or our machines) to accomplish learning, perception, interpretation, comprehension, etc., data had to be used effectively. We were looking for a way to do this, with hopes of using data in its most natural form possible. Conventional computers seemed to come up short in the crucial areas of representation and programming. Explicit, formal attempts to symbolize data could easily lead to the loss of important information. Knowledge bases were rigid and awkward to build. Conventional computers have much difficulty incorporating the concepts of perception, interpretation, learning and comprehension. Neural networks appeared to offer much promise toward accomplishing some of our goals.

However, before getting into neural networks, we were interested in taking a closer look at the characteristics of data itself. Some underlying components and features of data appeared to exist, but were not being utilized or even recognized well enough in attempts to automate information processing. In our investigation, we first analyzed signals in music to develop basic ideas on data and data analysis, and then extended some of the ideas to other forms of data. A kind of order or structure to data became apparent, which led to the development of the data spectrum. The spectrum was supposed to help us represent and be able to describe the underlying concepts involved in our analyses. Immediately the use of octaves, or intervals between two frequencies having a ratio of two to one, became a natural choice for a good way to describe data ranges. Harmony was also an obvious concept which needed to be considered. The ideas and concepts which resulted were all aimed at automating some of the processes involved in information processing. Neural network technology was conducive to the kinds of analyses we were interested in, even though many of the advantages of neural networks have still not been fully realized. Of course, we would take anything we could get from conventional computers, not the least of which was using them in our daily work.

5.2 The Data Spectrum

The data spectrum resulted from the desire to represent the natural order and structure of data in a graphical way, similar to the way the electromagnetic spectrum represents electromagnetic signals. Fundamental relationships, concepts and principles seemed to exist with respect to data, but they didn't seem to be characterized well enough. Our efforts provided us with a way to represent and discuss some of the issues involved in our research. Musical tones were analyzed using the computer to see how the frequencies of these audio signals were related. Raw data, in this case digitized music, contained tonal frequencies in the audio range of between 20 and 20K hertz. However, it was apparent that the frequencies at which we interpreted music, and also the *frequencies of* our *discussions* on it, were in a much lower range. Lower frequencies yet could be used to represent streams or sequences of data having longer duration, such as in songs, stories or motion pictures. Input signals to the brain (raw data) had the highest frequencies of all, while interpreting and thinking involved lower frequencies, and lowest of all were the frequencies of longer duration data sequences. A filtering process had to occur in order for data signals to be transformed from elements of higher frequencies to lower frequencies. Also, the complexity of these elements could vary at any given frequency. It was envisioned that the complexity of signals increased as one went from the core or center of the spectrum outward. The results are shown in figure 5-1.

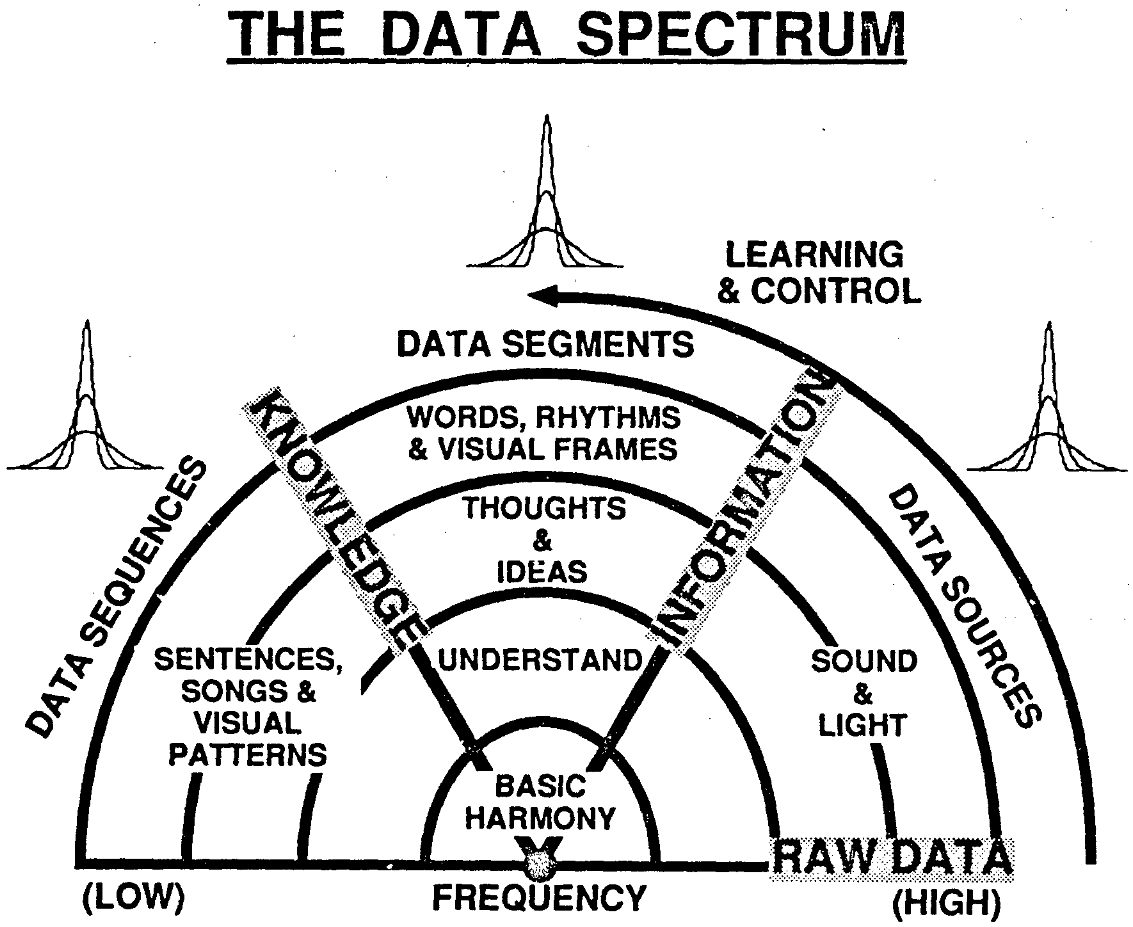


Figure 5-1. The Initial Data Spectrum. Data elements are shown as a function of frequency.

In considering many of the elements and concepts to be placed in the data spectrum, underlying issues concerning the actual nature of data came up. By this we mean that there seemed to be more to data than just its existence in raw form. Raw data is that which has not yet been organized. We all know of the form or state of data called information. This is data which has already been organized in some fashion, having some *form*. Knowledge can be considered yet another form of data. Naturally we wanted to map all forms of data onto the spectrum. Going from raw data to information to knowledge, the frequencies involved went from high to low. The sensitivity or timeliness of data also went from high to low going from raw data to knowledge. The amount of abstraction or complexity the data forms could take on went from low to high, with knowledge being the most abstract. Also, the amount of conscious control required by the receiving network to interpret the forms of data went from least to most control when going from raw data to knowledge. By now we see the shell of a spectrum which can be used to represent the complex nature of data, with its various forms and frequencies.

While all this was happening, other interesting characteristics of data were also being noticed. The existence of octaves, as well as the importance of harmony, were concepts which seemed to be related to the data spectrum. The use of the data spectrum was supposed to help define data relationships and enable us to get a handle on the enormous task of making sense of data. The framework that it provided seemed to enhance the concept of another framework we have come to call the octave rule. The existence of order in data, and a corresponding order and structure in data processing networks, were intriguing aspects to be considered in the development of automated information processing systems.

5.3 The Octave Rule

An octave in this sense is the smallest interpretable range possible in which a level of abstraction, or attention, can easily exist. The interval or range of an octave involves components of data whose frequencies, sizes, distances, etc., exist within a ratio of two to one. That is, data components within one octave have relative frequencies, distances, or sizes which are contained in a ratio of two to one. For example, if the ratio of frequencies of related data elements is more than two to one, than more than one octave is involved. The largest entity in an octave is by definition twice the smallest entity. By virtue of this small ratio, signals within one octave are already relatively close to each other. In concentrating on frequencies in our analysis, the actual *differences* in frequencies appeared to play a major role in the filtering and interpretation of data. By grouping data into manageable, understandable ranges, the brain is quickly and efficiently able to process the data. Without some kind of mechanism (such as the octave rule) to keep order, confusion or chaos would result. The existence of the octave rule appeared evident in music and in visual images, but further consideration indicated that the octave rule is an important concept in the effective management and control of many forms of data.

Images containing various shapes and sizes were generated using the computer to see how they may involve the octave rule. By focusing on various portions of images, the octave rule seemed to be followed by us as we focused and recognized objects. Data signals coming from the most recognizable and interpretable objects or entities seemed to fall within one octave. This was certainly a surprise to us. Also, the least pleasing or most difficult objects to look at covered more than one octave. Certain patterns appeared pleasing, while others appeared displeasing. Objects whose size conformed to the octave rule, using whatever units for size that were sensible, seemed to be focused on and recognized the soonest. Basic or primary colors were identified and selected more easily and quickly among many similar shades. Certain colors seemed to go well together while others did not. This implied that some fundamental, inherent characteristics of data existed, and that these characteristics were being exploited by the brain, under the heading of intelligent information processing.

The effects of the octave rule were of course very obvious in music. Pleasing sounds, as well as noisy or irritable sounds, could be created on the computer by combining musical tones inside and outside of octaves. It was also noticed that harmony and harmonics played an important role here, since you couldn't just mix any tones within an octave and get nice sounds. All of this implied that the brain, somewhere and somehow, had to group data into ranges in an effort to facilitate its interpretation. The fact that data was already grouped into pleasing, interpretable ranges in music was a case in point. Music has been recognized as a universally pleasing form of data. One hardly has to learn how to enjoy it. Music appreciation seems to be a built-in function of our brain. If it is, then a correlation has to exist between the data signals which comprise music and the nature and orderly operation of the brain. Even if the brain does not use the octave rule as envisioned above, if we could perform any of the brain's functions by making use of the octave rule, we would be making progress toward automating information processing.

Grouping or clustering of data is also part of the octave rule. Within one octave or range, an optimum number of data items or pieces of information exists, and it is a relatively small number. It is almost as if something was limiting the number of data items or elements within a range. This notion is related to the ability of interpreting only a certain amount of information at one time, after which point noise or confusion results. Usually the number of items or groups we can contain or think about is less than eight. Psychologists say seven. This would imply that as part of the attention or concentration process, something was limiting or filtering data, at or near its point of origin. As mentioned before, music provided a clear example of how information conformed to the octave concept. Also, musical chords are an example of how the octave rule involves only a small number of pleasing, interpretable items of data as a group. In any event, the octave rule seemed to be an important phenomenon, occurring in more than just music. The concept has provided insight into some of the mechanisms involved in data recognition and interpretation.

5.4 Harmony of Data

The concept of harmony was also related to the data spectrum and the octave rule, providing additional ways to help describe data interaction. In terms of data analysis, harmony involves the interference of many different kinds of signals. Interference involves the combination of many signals, with the results being determined by the laws of constructive and destructive interference. The amount of harmony in a network depends on how well the nature of the data conforms to the nature of the network's architecture. This means that data, and the network receiving and interpreting the data, should share some kind of order. Harmony is defined as a pleasing arrangement of data forms. In music, harmony helps determine how well the data signals sound. In images having colors, shapes, sizes, etc., harmony has to do with how recognizable or pleasing the images are to look at. For a system to be in harmony requires that its basic components be in agreement with each other. For a data analysis system, the signals which enter, get processed, stored, and communicated ought to be in some kind of harmony. This may not make much sense with respect to conventional computer systems, but it does when considering parallel forms of data, as in parallel processing.

The concept of harmony brings with it such terms as frequency distributions, resonant frequencies, signal means, harmonics, and overtones. The implications of incorporating these and other related features together in some kind of automated network would seem logical. However, the design of such a network or system is non-trivial, to say the least. The integration and coordination of parallel, simultaneous, complex signals in a useful, efficient manner is something machines will not be able to accomplish very well for many years. Yet we slowly approach that goal. Many complex techniques and tasks have already been automated, and new methods are constantly being developed. Hopefully some of the concepts and ideas which have been introduced here will contribute to the goal.

As an example of how important the concept of harmony may be, an analogy is made using the biological concept of homeostasis. The notion of homeostasis, and any stability resulting from its mechanisms, are believed to be essential for the existence and continuation of life [30]. The existence and use of stable signals or waveforms in the brain, in one form or another, should be a requirement for survival. The existence and use of intelligent waveforms, and the elimination of undesirable signals, would have to involve the concept of harmony in some way. It would follow that theories which explain why or how intelligent beings can even exist would have to include the concept of harmony as part of them. This indicates how important harmony may be in intelligent information processing systems. In this effort, we have probably raised more questions than answers by examining the nature of data and how signals get produced, transferred, and processed. Whatever the case, we hope to determine the roles complex yet orderly signals may play in the interpretation and communication of intelligible signals by starting with fundamental laws of physics and mathematics and describing some of the underlying functions of information processing. Perhaps then will we be able to automate some of these processes in an efficient manner.

5.5 Future Work

The work we have confronted in this effort represents the surface of several enormous tasks. We try to improve on reliability analysis in generic, far-reaching ways. We try to develop neural networks into useful, much-needed applications. And last but not least, we try to build intelligence into computers. Our work has only scratched the surface of these tasks. We have raised many unanswered questions. Along with what has been mentioned in this section and in the rest of this report, we have yet to consider many other important issues. Related areas such as time, probability, logic, non-linear functions, and adaptive control systems will have to be incorporated in the best of models. In order to do this, a better level of understanding of the issues involved will have to be reached. This will take time. We can build designs now, and our applications will be forced to evolve. But at the same time, we should have some of our sights set on the bigger picture of what we are trying to accomplish. This section has described some of our research, with its long term goals, on data concepts and data analysis methods. Much more research is needed, as is work in the many areas of application development.

While the emphasis of this work has been on electronic neural networks, it is very clear that the ultimate computing machine will be a combination of many different kinds of technologies, not just neural networks. We do not have to worry about the ultimate machine right now. At this point in time, we can only work on improving existing techniques. Conventional computers have much to offer, yet their limitations appear to be best overcome by the advantages offered by neural networks. But neural networks have disadvantages. And on it goes. Traditional AI continues to progress slowly. Many other technical areas will contribute to the technology of computing machines, including fuzzy logic, genetic algorithms, abductive reasoning and expert systems, not to mention the many hardware areas. Computing machines will definitely evolve.

The ultimate goal is to develop a system or network which performs useful data analysis functions in an automated fashion. This can be approached many different ways. The neural network approach envisions that mechanisms can be developed which perform functions somewhat analogous to those of the human brain. While it is believed that man-made neural networks will perform only some of the functions that living systems can perform, it is not envisioned that they should work in exactly the same way. At some level, the functionality will be similar, or perhaps the purpose will be similar, but the man-made systems will be much different from the living systems after which they are modeled. Two well known analogies which exhibit similarities and differences between nature and machine concern methods of travel. One analogy is between birds and planes, and the other compares legs with wheels. Each exemplifies similar function, or purpose, but the methods of operation are very different. Computers and brains are (and will be) very different in many respects. At the very least, they are made of different materials, resulting in very different chemical and physical processes. However, in a more practical sense, it is hoped that at least some of the overall functions of man-made information processing systems can be made similar to those of actual biological systems.

6.0 Perspective on What the Research Means

The basic research in this effort was aimed at providing a means for performing data analysis (eventually to be tailored for reliability analysis) in an automated fashion using neural network techniques. The desired neural network techniques do not exist, thus requiring research in this area. Since the means and mechanisms do not exist (in an automated fashion), we cannot exactly describe *how* they work. We can, however, introduce ways in which existing methods may be improved. This effort has involved investigating ways to perform automated data analysis using techniques modeled after the human brain (i.e. neural networks). While we cannot yet provide detailed descriptions on how the analyses should be performed, we can suggest how to begin modeling them. In this section we offer a larger perspective on what the research involves rather than details on how to solve particular problems. The insight gained from our research has enabled a much better understanding of the underlying processes. This section will describe how generic data analysis processes may work in humans, and how we may approach implementing some of these processes in computers using neural networks.

It was determined during the feasibility portion of this neural network effort to narrow the focus of the work to off-line software applications. Thus the work did not examine the neural network areas of biology, computer hardware, or real-time software applications. Admittedly, much research needs to be done in all of these areas, especially in the area concerning the reliability of neural network hardware. In any event, the work needed to be focused, and reliability theory and its basic analysis procedures were targeted first. In this focused area, under the heading of off-line software techniques, neural networks currently offer limited capability for performing such tasks as classification, modeling, optimization, and pattern recognition. These tasks align well with those performed in reliability analysis, which include allocation, correlation, diagnosis, evaluation, and prediction.

6.1 Automating Information Processing

To automate information processing, an appropriate model is required. As alluded to previously, many kinds of automated models and methods already exist, but none are powerful or versatile enough to enjoy widespread use. Put simply, the perfect model does not exist. This statement especially applies to the fledgling field of neural networks. Science and engineering have come a long way without the help of computers, providing theory and the means to explain and overcome many kinds of technical challenges. Mathematics has formed the foundation for these manual methods and techniques, with the use of math spreading across all technical disciplines.

Over the years, manual information processing has achieved very good results. In the whole scheme of things, manual or mental processes account for most of the processing done today, and humans will (arguably) always play a major role in information processing. Humans will definitely play a major role in *intelligent* information processing. However, with the advent of computers, more and more tasks are being automated, and this trend shows no signs of reversing.

Conventional computers have come a long way in a very short time, providing capabilities unmatched by humans (as machines should provide). The results of conventional computers have been useful, consistent, and accurate, obtained at very fast response times. Computers have become practical to design, build, use, and maintain. Their use is widespread, to say the least, but they do have severe limitations. Today's computers typically have only one central processing unit. This serial nature seriously limits the kinds of processing required for many complex tasks. Another drawback of conventional computers is their tedious, unforgiving requirement for programming. As good as computers are, they must be explicitly programmed to do exact sequences of operations, not allowing for any unforeseen deviations. As it turns out, most of the real world is far too complex to be described using explicit programming languages. Finally, conventional computers do not handle incomplete or imperfect data very well, which is a consequence of their explicit and rigid programming methods.

Research in learning has attempted to explain or better characterize the processes involved in information handling and data analysis. If these processes are ever to be automated, then limitations in conventional computers must be overcome. Also, the learning process must be better understood. To understand learning, its processes must be broken down into areas which can more readily be analyzed and investigated. The areas can all be perceived as involving data in some form or other. These areas include physical data sources, sensory input, filtering and focusing, feature extraction, harmony of data, association, comparison, interpretation, and memory, among others. Investigation of these concepts, and of the operations which must occur to allow them to exist, has indicated that a fundamental understanding of these processes is lacking. While this comes as no surprise, it does indicate that any attempts to automate these processes will come up short if not based on something solid. Work performed in this effort has attempted to examine fundamental processes, with the ultimate goal of being able to automate some of them in a useful, efficient fashion. The contributions of this research include a better understanding of information processing, ideas on how the learning process may work, and a realization that the advantages of both conventional computers and neural networks will have to be combined in future systems. Our approach to automate portions of information processing has emphasized neural networks.

6.2 The Learning Process

The most important and probably the least understood aspect of intelligent information processing is learning. A characterization of the learning process is proposed here which is based on frequency components of data. The research described in section 5 formed a starting point from which to investigate and explain the functions involved in learning. Section 3.2.1 described how learning is addressed in existing neural networks. In general, the entire learning process can be thought of as a transformation of data involving three forms: raw data, information and knowledge. This is reflected in the data spectrum of figure 5-1. Physically occurring raw data gets transformed into a transient state called information, and can then turn into a stable state called knowledge. Knowledge is what ultimately gets stored in memory, with information and raw data being intermediate, more time-sensitive phases which serve to feed the knowledge acquisition process. This perspective is by no means the only one possible. Especially confusing is the difference between knowledge and information. However, this perspective has proven useful in our investigation of data and learning. Frequency appears to be a crucial factor in the transformation of data in learning, and thus we have placed initial emphasis on the frequency aspects of data in our research.

To accomplish learning in the brain, data enters in a parallel fashion and gets filtered and focused according to its sensory type (e.g. audio, visual, etc.). Data gets further decomposed by the network in a process which uses relative differences in frequencies to filter and focus data into increasingly finer ranges. This process, proposed here in the form of the octave rule, is an attentional process which determines or extracts features within particular ranges as relevant or significant. The process helps allow data to be transformed from its raw state into information, and eventually into the form which gets stored in memory, called knowledge. All throughout the process, signals become associated with existing forms of data. These associations can be thought of as resulting from an interference process, having both positive and negative consequences. Depending on how well the data interferes, or *plays* together, the extent to which it is in harmony or agreement determines how well it can be interpreted or understood by the network. Again, differences in frequencies, among other things, between new signals and existing signals are used in the learning process. Eventually a corporate memory or knowledge base is accumulated within the network. This memory consists of many stored patterns which represent complex associations formed as data enters and passes through the network. Learning is the process which results from the many changes in data occurring in the network. Signal distributions are formed and stored in the network's memory in the form of connection weights. The distributions, which represent data associations, are composed of signals which have frequency, phase, and amplitude components.

It is ultimately some combination of these components which determines the characteristic state of the network at any given time. The *state* of the network dictates which types of signals can exist in the network, both in terms of being interpreted or comprehended as well as being stored or remembered.

This drastically simplified version of the learning process is enough to allow for a mechanism to begin to be developed which enables the implementation of these processes in an automated fashion, that is, in a machine. In so doing, one could extend the model further to accommodate processes which enable automated communication of data patterns. Communication is essentially the transfer of information, which is precisely what must happen for the learning process to occur. Useful, reliable forms of communication are so much a part of what we strive for in the way of information processing that it ought to be incorporated in future automated systems. In any event, the language of mathematics, and especially the areas of probability and statistics, will play a major role in realizing this. As mentioned previously, reliability science will benefit greatly from these accomplishments, since it has much to do with the issues of data analysis, probability and statistics, computerized techniques, and cause-and-effect relationships, all part of the automated learning process.

6.3 Harmony, Understanding, Thoughts, and Language

Data analysis and information processing occur in the human brain in such a way as to allow understanding, thought, and communication. The brain inputs signals using all of its senses, but seems to output using only two, verbal and body. Both of these forms of output can be considered forms of language. Language and communication are essential features in learning and intelligence, and they provide one of the few mechanisms by which to measure these processes. This section will discuss how harmony, understanding, thoughts, and language may be incorporated in a model which can be used to analyze data automatically. While this work has been theoretical rather than experimental, it serves as a useful perspective and also as a stepping stone for future research and applications.

If language is a key feature in intelligence, then it ought to be compatible with the more basic or primitive features of intelligence such as thought, understanding, and other cognitive processes (see figure 5-1). It is very difficult to evaluate language, and intelligence for that matter, in a strictly symbolic sense, as emphasized in traditional AI. Symbolism is by definition at least once removed from the concepts which it tries to represent. Granted, symbolism has its advantages, and is definitely useful in the end, but it is not altogether obvious that symbolism is what AI researchers are looking for as *a means to an end.* The gap of encoding intelligence is too large. A bottom-up approach based on the statistical and probabilistic nature of data does not preclude the use of a formal, symbolic language at some later point in a network. It is believed that attempts to represent information in it's most natural state, and to build up a hierarchical network based on concepts which are consistent and compatible with all forms of data, from the low end to the high end, will provide the insight needed as well as the means to automate information processing. The neural network approach embodies many of these features.

As data enters the brain, it gets converted from signals having higher frequencies to the much lower frequencies which the brain can ultimately store. Also, since data enters in a highly parallel fashion and is output in more of a serial fashion, much filtering must occur. This implies that the information content of data must get filtered in a very orderly fashion, keeping interesting features and ignoring extraneous ones, perhaps using the octave rule. For language and thought to occur, it must be in concert with the underlying operations of the brain. This just means that thought and language ought to be in harmony with the brain's basic operations. The work we have done implies that frequency is one of the main properties of the underlying operations of the brain. As a crude example, when something "makes sense", it may mean that the sensed signals (ideas, words, actions, etc.) are in harmony with existing signal distributions in the brain. In this context, harmony depends on the nature of data, as mentioned in section 5, and also on the existence of a network which makes use of the nature of data. We have to better realize how important order is in all forms of naturally occurring data, and we must develop mechanisms which allow many kinds of complex signals to be associated in constructive ways.

Interference is a key concept in harmony. When signals are added or mixed together, the constructive and destructive interference that takes place determines the resulting waveforms. Interference between signals appears to produce differences in signals which help the brain focus and concentrate, determine the relevance of signals, establish priorities, help make decisions, and know what to ignore. In terms of physics, basic laws and concepts must be followed for a stable network to exist, and a description of the network should be possible using the language of mathematics. Many other disciplines and skills are needed as well. An interdisciplinary approach is needed in order to engineer fundamental concepts into useful systems.

Our work has centered around establishing a framework or foundation on which to develop and build intelligent machines. This includes automating methods which incorporate learning, understanding, and communication. Definitions follow which are admittedly and necessarily general, given the framework of the research and the abstraction of the topic. Intelligence is defined as the ability to translate information into knowledge. This includes the ability to learn and effectively apply knowledge in a changing environment. Learning is defined as acquiring knowledge, incorporating the concepts of change and purpose. Knowledge is defined as familiarity gained through experience or association, and also as that body of information which results from an experience. Understanding is defined as the ability to interpret, accept as plausible, grasp the significance of, or the capacity to make generalizations. Thought is the process which creates and uses waveforms, which includes the characteristics of priority, relevance and significance of information. Communication is defined as the transfer of information. All of these concepts have common threads, one being *information*. It appears as though a model can be formed which hinges upon the association of these concepts. It should be noted that representation is a major issue here. Each of the concepts mentioned above can and do have different meanings, and the method of representing them is always an important concern.

6.4 Reliable Communication - Getting the Point Across

It was stated earlier that useful, reliable forms of communication are extremely important in information processing. For information processing to be considered *intelligent*, however, something more than communication, more than the transfer of information, is required. It is necessary to be able to represent and convey certain significant data characteristics inherent in intelligent communication. These characteristics can be described as data descriptors or *statistics*. With communication being the transfer of information, intelligent communication is considered to involve *getting the point across*. The distinction here is noted in the mathematical analog or equivalent of *the point*. The *point*, in a very simple sense, is considered the arithmetic *mean*. Communication signals can be represented as data distributions, with each of them having a reference or *mean*. The *mean* represents a significant component or characteristic of a distribution. Intelligent communication relies on such components - they can be more significant than raw data itself. The purpose of intelligent communication would then be to get the point, or mean, across. Other statistical terms and concepts can also be used to represent and process intelligent communication. Of course, intelligent communication usually involves many signals, but for now we assume that they can be combined into a small number of significant data distributions over a certain time interval, perhaps at the expense of changing levels of abstraction (octaves). Basic components of intelligent communication signals do exist, and their mathematical representation is what we are after.

Communication consists of signals which must represent and convey the many features and aspects of information inherent in the communication. The brain, upon receiving and recognizing these signals, must form useful data associations or relationships. These associations must be an accurate and appropriate representation of the features and characteristics of the input signals in order for communication to be effective. Data features entering the network will either form new connections (associations), modify existing ones, or be ignored. Data occurrences, approximations, and averages can all be represented using frequency distributions. It is believed that the communication process, which underlies information processing and understanding, can be represented in terms of physical signals and mathematical functions. Probability and statistics especially can and should be used more effectively in the development of information processing systems.

During communication, the data form called information is transferred from some source to some destination. The actual signals being transferred can be expressed as sine waves originating from some physical source. The combinations and common occurrences of these signals can be represented using frequency distributions. For large amounts of data, normal or gaussian distributions can be used to approximate many of the signal associations which are formed and used in communication and learning. While it is extremely difficult if not impossible to know exactly what is going on in the brain during learning, we can use approximations and averages to represent the probabilistic and statistical nature of data signals involved in these processes. In terms of implementing some of these processes in machines, we can use the characteristics of the normal curve to describe some of the functions which occur. By averaging random phenomenon over many observations, we can analyze, predict, and in general draw conclusions from data.

The communication process always involves the transfer of data in some form or other. Groups of data can collectively be called messages. Since a message can consist of many pieces of data, it becomes important to be able to identify and represent the main point of a message. Phrases such as "get to the point!" and "what is your point?" emphasize the existence of main themes of messages. As already mentioned, the *point* can be considered the arithmetic mean of a signal distribution. The mean could be used to identify the *main point* of a signal distribution, representing the combination of many data components which constitute that signal. When many signals are involved, the *main point* would be some arithmetic mean or average of many signal distributions. We should be able to represent and describe the signals mathematically. The identification of the main points in communication, as well as the data components which reinforce them (i.e., other *statistics*), would lead to *understanding*. Understanding is ideally envisioned as occurring when the arithmetic means of (input) signals get aligned, to some degree, with those of existing signals. While these statements are very simplistic, the underlying principles cannot be ignored. We try not to build exact copies of biological systems, but to build useful models. Certainly communication involves more than just getting the point, or mean, across. Simple concepts are good to start with, provided they are not too simple.

Statistics provides many terms and concepts which can be used to describe data, and thus can be very useful in describing information processing. Descriptive and inferential statistics, probability, combinatorics, samples and populations, dispersion, correlation, estimation, mean, variance, standard deviation, confidence intervals, hypothesis testing, regression analysis, and the central limit theorem are just a few of the important concepts and terms [16] which must be applied to the information processing domain. Unfortunately, the information processing domain has yet to be refined, if it is even defined. The preliminary concepts and ideas presented here (or anywhere for that matter) on automating information processing should eventually be subject to all the rigors of mathematics and appropriately tested.

The communication process can also be described in terms of a cause and effect relationship. The cause is the source or originator of communication. The source conveys some message or entity to some destination. The communication may be one-way, or perhaps caused by some event in nature which may not even be considered intelligent. However, the issue then would become one of intelligent interpretation of input signals rather than of intelligent communication. In any event, signals get physically created or produced by some source. The destination then receives the message or entity and incorporates it into its network. Of course the extent of interpretation is proportional to the amount comprehended or learned, which has to do with how effective the communication or signal transfer was. The result, or effect, is a physical change in the destination network. The change in the context of neural networks would most likely take place in the form of modified connection weights.

6.5 Data Analysis - Making Sense of It All

Data analysis and information processing are terms which have been used somewhat interchangeably in this report. No attempt has been made to associate them with the meanings suggested in section 5 for the root words *data* and *information*. The difference there between data and information was described in terms of levels of sophistication, with raw data being the lowest form. The assumption overriding all of this discussion is that communication is essential to data analysis and information processing. Whether done by man or machine, and whether performing analysis, computation, processing, inputting or outputting, communication has to be involved. As already mentioned, this discussion involves the development of intelligent systems. With communication as an overall requirement, a characterization of the communication process is necessary. This effort has attempted to investigate and establish fundamental concepts involved in intelligent communication, certainly part of the big picture. The work has involved analysis of neural networks and other computer techniques, and has looked at similar human cognitive processes for insight. This work has been conceptual and abstract, attempting to characterize very complex processes. Given this, it is very difficult to prove many of the statements made here. However, it is believed that attempts to substantiate or disprove these claims can and should be made in the future, and will prove very useful.

Another important issue needs to be brought up now. The issue has to do with the overall purpose of data analysis, with living systems providing important precedents. The issue is very debatable, but it is believed that *one of the main purposes of intelligent information processing and data analysis systems is to provide the mechanisms and abilities needed to make good decisions.* The survival of intelligent beings depends to a large part on their decision-making abilities. Good decisions require knowledge of relevant factors and a good understanding of what each of the factors mean. The understanding does not have to be very advanced, but the consequences of making a decision must be clear to be able to learn from the decision. Along with typical definitions of what a decision is, decision mechanisms involve such issues as reference signals, sums and differences, variations, priority schemes, thresholds, trade-offs, uncertainty and change. To make so-called "good" decisions, many complicated processes must occur. To automate some of these functions, not only do we have to understand the processes, but we also have to be able to implement them in some kind of network. This is no easy task.

The network of choice will ultimately be some kind of computer, but it will have to incorporate more than conventional hardware and software techniques. Based on the state-of-the-art of conventional computers and our understanding of intelligent information processing, today's computers do not have what it takes to handle the concepts of intelligence and learning. Serial architectures and rigid programming have left too much to be desired as far as computers are concerned. Lack of understanding of what intelligent processes really involve has eluded researchers thus far, leaving areas such as traditional artificial intelligence unable to capture the essence of intelligence. Traditional AI has provided many inroads, however, such as in the areas of knowledge based systems and automated reasoning, but something fundamental still appears to be missing. If we knew how intelligence worked, we could program it on our favorite computer. However, we do not. Other approaches will have to be considered. Models need to be based on fundamental concepts rather than abstract ones. We suggest first investigating the physics of data. What seems to be missing from all of information processing is a fundamental understanding of the nature of data. Also, basic concepts in mathematics, such as probability theory and statistics, should be used more effectively. Decision theory may be a good place to start. Perhaps existing theories need to be refined, or new ones proposed to help researchers overcome existing computational bottlenecks. Something has to change to make way for better automated techniques. The approach of this effort has been to consider the big picture (top-down) of automating information processing, and then to construct a model based on fundamental principles (bottom-up); that is, to understand the underlying concepts of data analysis and information processing, and then to develop techniques and models which incorporate these processes in effective ways.

To summarize what has been discussed in this section, the perspective is that data analysis and information processing can be performed in machines using models which perform functions analogous to those in the human brain. The purpose, arguably, is to enable intelligent decisions. Intelligence requires understanding, which requires communication, which requires language. Understanding involves the ability to learn and to generalize. All of these processes must be realized and accomplished in some kind of network which exploits the inherent components of data. Frequency components have been emphasized in this work. Signals containing high frequency components enter the brain in a parallel fashion. Data gets filtered and focused into various ranges or levels (octaves), resulting in lower and lower frequencies. Harmonic signals get formed as a result of signal interferences in the network. Signals with the strongest harmonics are those which best represent the nature of the data and conform most to the structure of the network. Resulting signals can be used by the network to learn, and can be stored as knowledge in the form of connection weights (memory). The weights represent various forms and kinds of data associations, and can be output using some kind of language, resulting in communication, which keeps the process going.

For intelligent information processing to exist within a network, the network must conform to laws of physics. Mathematics can be used to describe the internal processes and functions. Probability and statistics will provide much toward this end. Probability provides concepts which represent the true random nature of data, while statistics provides useful and powerful methods for describing the nature of data. Reliability analysis will benefit greatly from the resulting developments, and it will also provide as a contribution many of the characterizations and methods which are already part of its science. Concepts and terms from reliability science can be used to help describe and characterize many of the forms and uses of data in automated information processing, and developments which result from automating information processing can be used to improve reliability analysis itself. The fundamental links which exist between information processing and reliability theory will thus benefit both areas. It is our intent to tie together fundamental concepts in physics, mathematics, reliability, computers, engineering, and cognitive science, among other disciplines, in effort to automate functions which would be desirable and useful in future Air Force systems.

7.0 Statistical Neural Network - A Prototype Application

Many current neural network applications provide unique solutions to problems which are too abstract in nature to be programmed by conventional methods. The electronic versions of neurons and their connections can provide parallel processing, adaptive learning, and other features which today's computers cannot replicate. However, along with these abilities, neural networks bring new problems and complexities which must be solved.

One of the major hurdles in developing a neural network application is choosing or establishing a suitable architecture. Since the neural network architecture must handle data effectively and efficiently, data statistics are a good starting point when addressing architecture considerations. To demonstrate how statistics can be used to help design a neural network architecture, this section will describe the design process of the Statistical Neural Network, provide a step by step description of the network's operation, and give an example of the Statistical Neural Network in action. The significance of this application is to show how statistics can be used to describe natural tendencies in data, which can lead to more efficient neural network designs and data analysis capabilities.

7.1 Statistical Network Design

The Statistical Neural Network uses statistical features of data to aid in the design of the network's architecture. The more available and representative the data is, the better the results. Various data descriptors can be used in the design of a network, such as:

- **location** (e.g., mean, median, mode, etc.)
- **dispersion** (e.g., range, variance, standard deviation, coefficient of variance, etc.)
- **correlation** (e.g., covariance, correlation coefficient, linear regression, etc.)

For this application, data was generated by computer with the aid of the mathematical software package called Mathematica [31]. Figures 7-1 through 7-4 represent the data sets generated and used for this application. The data in these graphs do not have any units attached to them yet, representing four generic data sets labelled data1 through data4, respectively. Units for the four data sets used in this particular application will be assigned later.

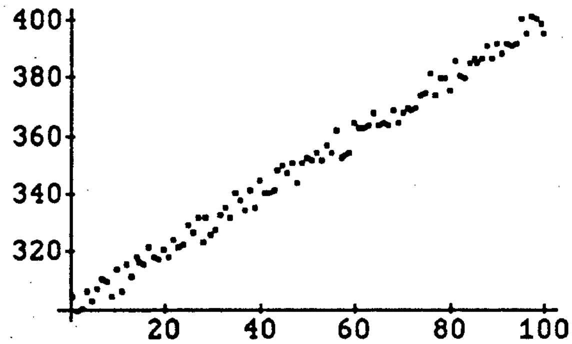


Figure 7-1 Data1 set of data generated

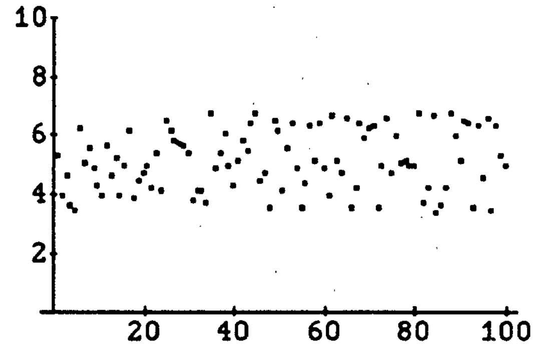


Figure 7-2 Data2 set of data generated

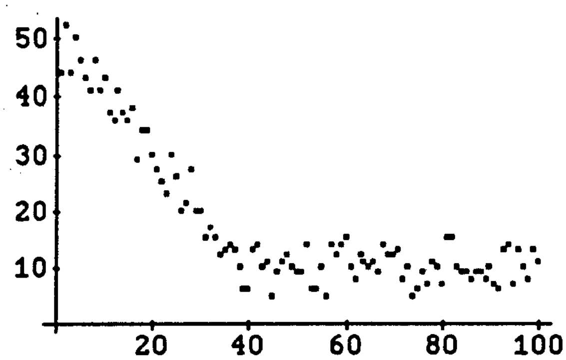


Figure 7-3 Data3 set of data generated

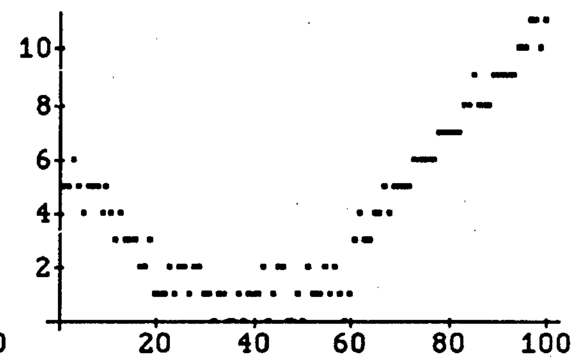


Figure 7-4 Data4 set of data generated

The theory behind the Statistical Neural Network is that it conforms its architecture to data through the calculation and use of important data descriptors. Many kinds of data descriptors exist, and there are many ways to combine them [16]. To illustrate how statistics can be used in the design of a network architecture, the following basic descriptors were used in this application:

- **mean**
- **range**
- **correlation coefficient**
- **linear regression line**
- **mean square error**

Before calculating values for these descriptors, it is necessary to preprocess the data by grouping it into corresponding sets. The data is listed below in tabular format, consisting of one hundred data samples, in sets of four:

data = {{304, 5.31837, 44, 5}, {299, 3.91846, 52, 5}, {300, 4.62778, 44, 6}, {306, 3.6149, 50, 5}, {302, 3.38225, 46, 4}, {307, 6.20638, 43, 5}, {310, 5.00446, 41, 5}, {309, 5.54532, 46, 5}, {304, 4.846, 41, 4}, {313, 4.2878, 43, 5}, {306, 3.93481, 37, 4}, {315, 5.58198, 36, 3}, {311, 4.59175, 41, 4}, {318, 5.18741, 37, 3}, {316, 3.94637, 36, 3}, {315, 4.91556, 38, 3}, {321, 6.09458, 29, 2}, {318, 3.8803, 34, 2}, {317, 4.46967, 34, 3}, {320, 4.65441, 30, 1}, {318, 4.93863, 27, 1}, {324, 4.20427, 25, 1}, {321, 5.34331, 23, 2}, {322, 4.07454, 30, 1}, {329, 6.49557, 26, 2}, {326, 6.14121, 20, 2}, {331, 5.77097, 21, 1}, {323, 5.68284, 27, 2}, {331, 5.61888, 20, 2}, {325, 5.40303, 20, 1}, {327, 3.79244, 15, 1}, {332, 4.07016, 17, 0}, {335, 4.13761, 15, 1}, {331, 3.66273, 12, 1}, {340, 6.67676, 13, 0}, {337, 4.83665, 14, 0}, {334, 5.3484, 13, 1}, {341, 6.07406, 10, 0}, {335, 4.90646, 6, 1}, {344, 4.22754, 6, 1}, {340, 5.14997, 13, 1}, {340, 5.75343, 14, 2}, {341, 5.41901, 10, 0}, {348, 6.39532, 11, 1}, {349, 6.68047, 5, 2}, {347, 4.44835, 9, 2}, {350, 4.70566, 11, 0}, {343, 3.46611, 12, 0}, {350, 6.43853, 10, 1}, {352, 6.08991, 9, 0}, {351, 4.05826, 9, 2}, {354, 5.56266, 14, 1}, {351, 6.37007, 6, 1}, {356, 4.87563, 6, 1}, {354, 3.53548, 10, 2}, {361, 4.37084, 5, 1}, {352, 6.25291, 14, 2}, {353, 5.14325, 12, 1}, {354, 6.35959, 14, 0}, {364, 4.87547, 15, 1}, {362, 3.93456, 10, 3}, {362, 6.61371, 8, 4}, {363, 5.07228, 12, 3}, {367, 4.64878, 11, 3}, {363, 6.58352, 10, 4}, {364, 3.50017, 11, 4}, {363, 4.17494, 9, 5}, {368, 6.34582, 14, 4}, {364, 5.86369, 12, 5}, {367, 6.19878, 12, 5}, {369, 6.26682, 13, 5}, {368, 3.51247, 8, 5}, {369, 4.96642, 10, 6}, {373, 6.55415, 5, 6}, {374, 4.67431, 6, 6}, {381, 5.99145, 9, 6}, {373, 5.03789, 7, 6}, {379, 5.10556, 11, 7}, {379, 4.92649, 10, 7}, {375, 4.91083, 7, 7}, {385, 6.69842, 15, 7}, {380, 3.67288, 15, 7}, {379, 4.20384, 10, 8}, {384, 6.62349, 9, 8}, {386, 3.37296, 9, 9}, {384, 3.55709, 8, 8}, {386, 4.2112, 9, 8}, {390, 6.68575, 9, 8}, {386, 5.98952, 8, 9}, {391, 5.12559, 10, 9}, {388, 6.4455, 7, 9}, {391, 6.34277, 6, 9}, {390, 3.5013, 13, 9}, {391, 6.28877, 14, 10}, {400, 4.51984, 7, 10}, {395, 6.50837, 13, 10}, {401, 3.46258, 10, 11}, {400, 6.26843, 8, 11}, {398, 5.26257, 13, 10}, {395, 4.89994, 11, 11}}

Next, the data sets are divided into four parts. For this example, let us assign the four data sets to represent the parameters temperature, vibration, humidity, and number of failures, respectively. Each of the four data parameters are listed separately below:

data1 = temperature = {304, 299, 300, 306, 302, 307, 310, 309, 304, 313, 306, 315, 311, 318, 316, 315, 321, 318, 317, 320, 318, 324, 321, 322, 329, 326, 331, 323, 331, 325, 327, 332, 335, 331, 340, 337, 334, 341, 335, 344, 340, 340, 341, 348, 349, 347, 350, 343, 350, 352, 351, 354, 351, 356, 354, 361, 352, 353, 354, 364, 362, 362, 363, 367, 363, 364, 363, 368, 364, 367, 369, 368, 369, 373, 374, 381, 373, 379, 379, 375, 385, 380, 379, 384, 386, 384, 386, 390, 386, 391, 388, 391, 390, 391, 400, 395, 401, 400, 398, 395}

data2 = vibration = {5.31837, 3.91846, 4.62778, 3.6149, 3.38225, 6.20638, 5.00446, 5.54532, 4.846, 4.2878, 3.93481, 5.58198, 4.59175, 5.18741, 3.94637, 4.91556, 6.09458, 3.8803, 4.46967, 4.65441, 4.93863, 4.20427, 5.34331, 4.07454, 6.49557, 6.14121, 5.77097, 5.68284, 5.61888, 5.40303, 3.79244, 4.07016, 4.13761, 3.66273, 6.67676, 4.83665, 5.3484, 6.07406, 4.90646, 4.22754, 5.14997, 5.75343, 5.41901, 6.39532, 6.68047, 4.44835, 4.70566, 3.46611, 6.43853, 6.08991, 4.05826, 5.56266, 6.37007, 4.87563, 3.53548, 4.37084, 6.25291, 5.14325, 6.35959, 4.87547, 3.93456, 6.61371, 5.07228, 4.64878, 6.58352, 3.50017, 4.17494, 6.34582, 5.86369, 6.19878, 6.26682, 3.51247, 4.96642, 6.55415, 4.67431, 5.99145, 5.03789, 5.10556, 4.92649, 4.91083, 6.69842, 3.67288, 4.20384, 6.62349, 3.37296, 3.55709, 4.2112, 6.68575, 5.98952, 5.12559, 6.4455, 6.34277, 3.5013, 6.28877, 4.51984, 6.50837, 3.46258, 6.26843, 5.26257, 4.89994}

data3 = humidity = {44, 52, 44, 50, 46, 43, 41, 46, 41, 43, 37, 36, 41, 37, 36, 38, 29, 34, 34, 30, 27, 25, 23, 30, 26, 20, 21, 27, 20, 20, 15, 17, 15, 12, 13, 14, 13, 10, 6, 6, 13, 14, 10, 11, 5, 9, 11, 12, 10, 9, 9, 14, 6, 6, 10, 5, 14, 12, 14, 15, 10, 8, 12, 11, 10, 11, 9, 14, 12, 12, 13, 8, 10, 5, 6, 9, 7, 11, 10, 7, 15, 15, 10, 9, 9, 8, 9, 9, 8, 10, 7, 6, 13, 14, 7, 13, 10, 8, 13, 11}

data4 = number of failures = {5, 5, 6, 5, 4, 5, 5, 5, 4, 5, 4, 3, 4, 3, 3, 3, 2, 2, 3, 1, 1, 1, 2, 1, 2, 2, 1, 2, 2, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 2, 0, 1, 2, 2, 0, 0, 1, 0, 2, 1, 1, 1, 2, 1, 2, 1, 0, 1, 3, 4, 3, 3, 4, 4, 5, 4, 5, 5, 5, 5, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 8, 8, 9, 8, 8, 8, 9, 9, 9, 9, 9, 10, 10, 10, 11, 11, 10, 11}

Data descriptors are then calculated for each range harmonic, from the first up through the seventh harmonic. Range harmonic is a term used to describe a portion of the data set. For example, the first harmonic represents the entire data set; the second harmonic, first sector represents the first half of the data set; the second harmonic, second sector represents the second half of the data set, and so on. The number of range harmonics can vary, depending on the particular application. For this case, seven appeared to be a good number to start with. The data was presented as it appears in Figures 7-1 through 7-4, but since these descriptors are dimensionless (independent of any units), the data can be used in any sequence.

The formulas used to calculate values for the various data descriptors in the first range harmonic are given below. The variables used in the formulas are abbreviated as follows:

- **data1** - temperature
- **data2** - vibration
- **data3** - humidity
- **data4** - number of failures
- **mean** - mean
- **var** - variance
- **range** - range
- **MSE** - mean square error
- **b1** - constant
- **b0** - constant
- **s** - covariance
- **r** - correlation coefficient
- **norm** - normal coefficient
- **yhat = b0 + b1(x)** - linear regression line

The numbers in the variables indicate which data set was used (e.g., **r12** is the correlation coefficient between data1 and data2; norm34 is the normal coefficient for data3 used to find data4). The formulas are illustrated in Mathematica format as follows:

data1mean = N[(1/n)Sum[data1[[i]], {i, n}]] = 350.1

data1var = N[(1/(n-1))Sum[(data1[[i]] - data1mean)^2, {i, n}]] = 831.768

data1range = Max[data1] - Min[data1] = 102

data2mean = N[(1/n)Sum[data2[[i]], {i, n}]] = 5.09889

data2var = N[(1/(n-1))Sum[(data2[[i]] - data2mean)^2, {i, n}]] = 1.02994

data2range = Max[data2] - Min[data2] = 3.32546

data3mean = N[(1/n)Sum[data3[[i]], {i, n}]] = 17.7

data3var = N[(1/(n-1))Sum[(data3[[i]] - data3mean)^2, {i, n}]] = 157.283

data3range = Max[data3] - Min[data3] = 47

data4mean = N[(1/n)Sum[data4[[i]], {i, n}]] = 4.01

data4var = N[(1/(n-1))Sum[(data4[[i]] - data4mean)^2, {i, n}]] = 10.0504

data4range = Max[data4] - Min[data4] = 11

b1 = (Sum[data2[[i]]data1[[i]], {i, n}] - (Sum[data2[[i]],
{i, n}]Sum[data1[[i]], {i, n}])/n)/(Sum[data1[[i]]^2,
{i, n}] - n(data1mean^2)) = 0.00614082

b0 = data2mean - (data1mean)b1 = 2.94899

yhat12 = b0 + b1(x) = 2.94899 + 0.00614082 x

MSE12 = (1/(n-2))Sum[(data2[[i]] - (b0 + (b1)data1[[i]]))^2, {i, n}] = 1.00876

b1 = (Sum[data3[[i]]data1[[i]], {i, n}] - (Sum[data3[[i]],
{i, n}]Sum[data1[[i]], {i, n}])/n)/(Sum[data1[[i]]^2,
{i, n}] - n(data1mean^2)) = -0.346894

b0 = data3mean - (data1mean)b1 = 139.148

yhat13 = b0 + b1(x) = 139.148 - 0.346894 x

MSE13 = (1/(n-2))Sum[(data3[[i]] - (b0 + (b1)data1[[i]]))^2, {i, n}] = 57.7752

b1 = (Sum[data4[[i]]data1[[i]], {i, n}] - (Sum[data4[[i]],
{i, n}]Sum[data1[[i]], {i, n}])/n)/(Sum[data1[[i]]^2,
{i, n}] - n(data1mean^2)) = 0.0700091

b0 = data4mean - (data1mean)b1 = -20.5002

yhat14 = b0 + b1(x) = -20.5002 + 0.0700091 x

MSE14 = (1/(n-2))Sum[(data4[[i]] - (b0 + (b1)data1[[i]]))^2, {i, n}] = 6.03464

b1 = (Sum[data3[[i]]data2[[i]], {i, n}] - (Sum[data3[[i]],
{i, n}]Sum[data2[[i]], {i, n}])/n)/(Sum[data2[[i]]^2,
{i, n}] - n(data2mean^2)) = -2.43737

b0 = data3mean - (data2mean)b1 = 30.1279

yhat23 = b0 + b1(x) = 30.1279 - 2.43737 x

MSE23 = (1/(n-2))Sum[(data3[[i]] - (b0 + (b1)data2[[i]]))^2, {i, n}] = 152.707

b1 = (Sum[data4[[i]]data2[[i]], {i, n}] - (Sum[data4[[i]],
{i, n}]Sum[data2[[i]], {i, n}])/n)/(Sum[data2[[i]]^2,
{i, n}] - n(data2mean^2)) = 0.0582553

b0 = data4mean - (data2mean)b1 = 3.71296

yhat24 = b0 + b1(x) = 3.71296 + 0.0582553 x

MSE24 = (1/(n-2))Sum[(data4[[i]] - (b0 + (b1)data2[[i]]))^2, {i, n}] = 10.1494

b1 = (Sum[data4[[i]]data3[[i]], {i, n}] - (Sum[data4[[i]],
{i, n}]Sum[data3[[i]], {i, n}])/n)/(Sum[data3[[i]]^2,
{i, n}] - n(data3mean^2)) = -0.0288164

b0 = data4mean - (data3mean)b1 = 4.52005

yhat34 = b0 + b1(x) = 4.52005 - 0.0288164 x

MSE34 = (1/(n-2))Sum[(data4[[i]] - (b0 + (b1)data3[[i]]))^2, {i, n}] = 10.021

s12 = N[(1/(n-1))Sum[(data1[[i]] - data1mean)
(data2[[i]] - data2mean), {i, n}]] = 5.10773

s13 = N[(1/(n-1))Sum[(data1[[i]] - data1mean)
(data3[[i]] - data3mean), {i, n}]] = -288.535

s14 = N[(1/(n-1))Sum[(data1[[i]] - data1mean)
(data4[[i]] - data4mean), {i, n}]] = 58.2313

s23 = N[(1/(n-1))Sum[(data2[[i]] - data2mean)
(data3[[i]] - data3mean), {i, n}]] = -2.51034

s24 = N[(1/(n-1))Sum[(data2[[i]] - data2mean)
(data4[[i]] - data4mean), {i, n}]] = 0.0599993

s34 = N[(1/(n-1))Sum[(data3[[i]] - data3mean)
(data4[[i]] - data4mean), {i, n}]] = -4.53232

r12 = s12/(Sqrt[data1var]Sqrt[data2var]) = 0.174511

r13 = s13/(Sqrt[data1var]Sqrt[data3var]) = -0.797733

r14 = s14/(Sqrt[data1var]Sqrt[data4var]) = 0.636889

r23 = s23/(Sqrt[data2var]Sqrt[data3var]) = -0.197236

r24 = s24/(Sqrt[data2var]Sqrt[data4var]) = 0.0186487

r34 = s34/(Sqrt[data3var]Sqrt[data4var]) = -0.113996

norm12 = (r12)constant12 = 0.0203322

norm13 = (r13)constant13 = -2.14498

norm14 = (r14)constant14 = 0.313489

norm23 = (r23)constant23 = -9.27024

norm24 = (r24)constant24 = 0.201144

norm34 = (r34)constant34 = -0.100133

The number of correlation coefficients needed for the Statistical Neural Network is equal to the combination of the number of data parameters (e.g., four) taken two at a time (i.e., 4!/2!2!). For the normal coefficients, order matters, so their number equals the number of permutations of data parameters taken two at a time (i.e., 4!/2!). Therefore, for each range harmonic, there are 6 correlation coefficients and 12 normal coefficients. In addition, there are 12 linear regression lines for each range harmonic. A linear regression line provides the user with a way to estimate one parameter given another. For example, the linear regression line "1 to 2" can be used to predict data parameter 2 given data parameter 1. Figures 7-5 through 7-10 below represent graphs of six linear regression lines: 1 to 2, 1 to 3, 1 to 4, 2 to 3, 2 to 4 and 3 to 4, respectively. The other six related regression lines (2 to 1, 3 to 1, 4 to 1, 3 to 2, 4 to 2 and 4 to 3) could be plotted by inverting the axes. It is apparent that these lines do not fit the initial data set very well. This is why it is necessary to look at an adequate number of range harmonics in order to get good results.

After performing the above calculations for all seven range harmonics (28 in all), then the best results are used to help build the network. The key factors used to determine which results are best are the normal coefficient and the correlation coefficient.

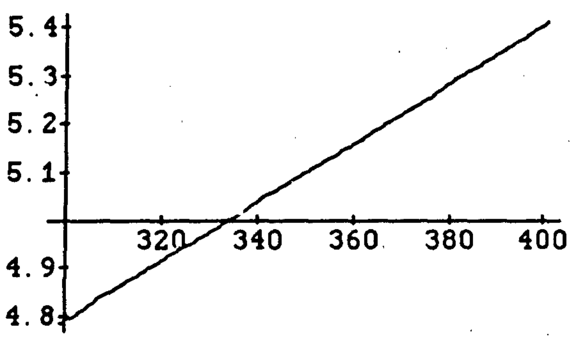


Figure 7-5 Linear regression line (1 to 2)

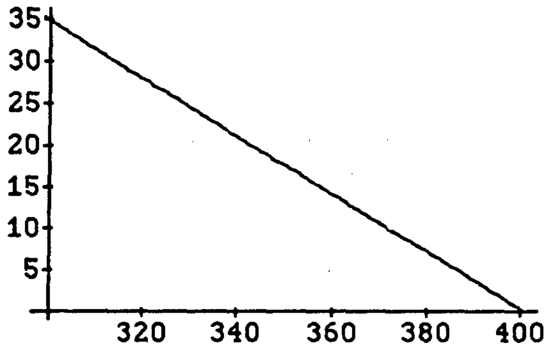


Figure 7-6 Linear regression line (1 to 3)

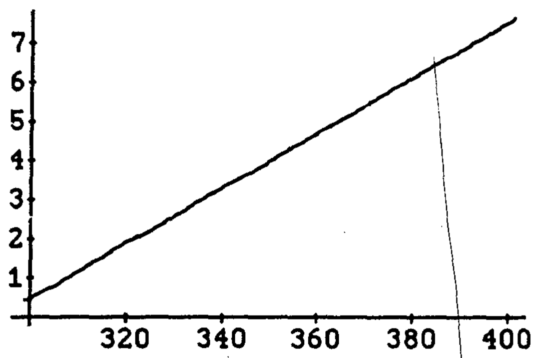


Figure 7-7 Linear regression line (1 to 4)

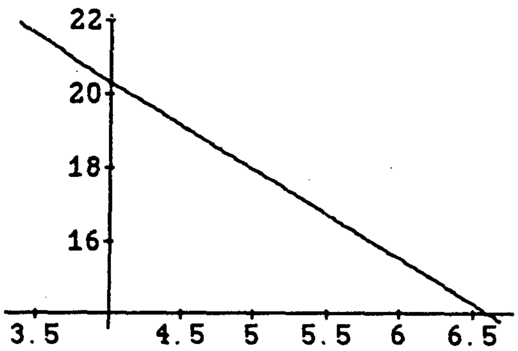


Figure 7-8 Linear regression line (2 to 3)

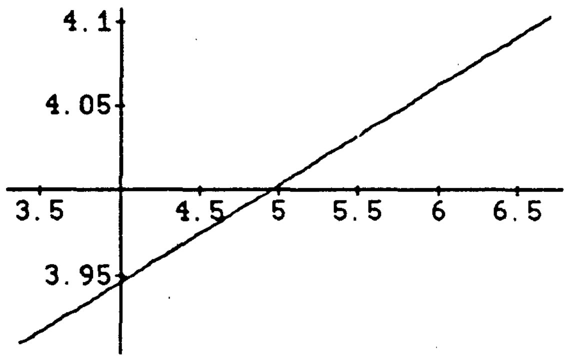


Figure 7-9 Linear regression line (2 to 4)

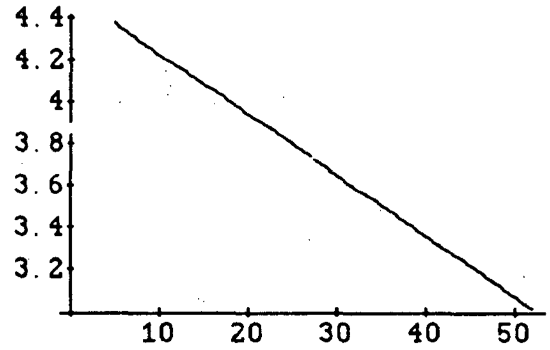


Figure 7-10 Linear regression line (3 to 4)

The normal coefficient equals the range of the data set divided by the square root of the mean squared error (MSE). This scalar number is normalized between 0 and 1, and then multiplied by the correlation coefficient. It is named the normal coefficient because it normalizes two important factors (range and MSE) used to determine the best range harmonic for each particular range. The correlation coefficient is a number between -1 and 1 and represents the amount of linear correlation between two data sets. The closer the number is to 1, the greater the degree of linear correlation (as in Figures 7-5, 7-7 and 7-9 above); the closer the number is to -1, the greater the degree of inverse linear correlation (as in Figures 7-6, 7-8 and 7-10 above); the closer the number is to zero, the lesser the degree of linear correlation.

Using normal coefficients, correlation coefficients and range harmonics, the network can now be constructed. Figures 7-11 through 7-16 are bar graphs representing all the normal coefficients (entitled "Normal (# to All)") and correlation coefficients (entitled "Linear Relationship") for all seven range harmonics. To make it somewhat easier to read, all the solid bars represent odd range harmonics, while all the diagonally slashed bars represent even range harmonics. Figures 7-11 through 7-14 are used to determine the number of nodes in the first hidden layer (priority rating) while Figures 7-15 and 7-16 provide a confidence rating of linearity for the results of the network.

The network constructed from this data is represented in Figure 7-17. Starting from the top and working down, the first layer consists of four input nodes. These nodes take in values for each of the four input parameters. The next layer, the first hidden layer, has all the nodes necessary to represent all combinations of four data sets taken two at a time, with all ranges included within each combination. These nodes are created by taking the greatest absolute values

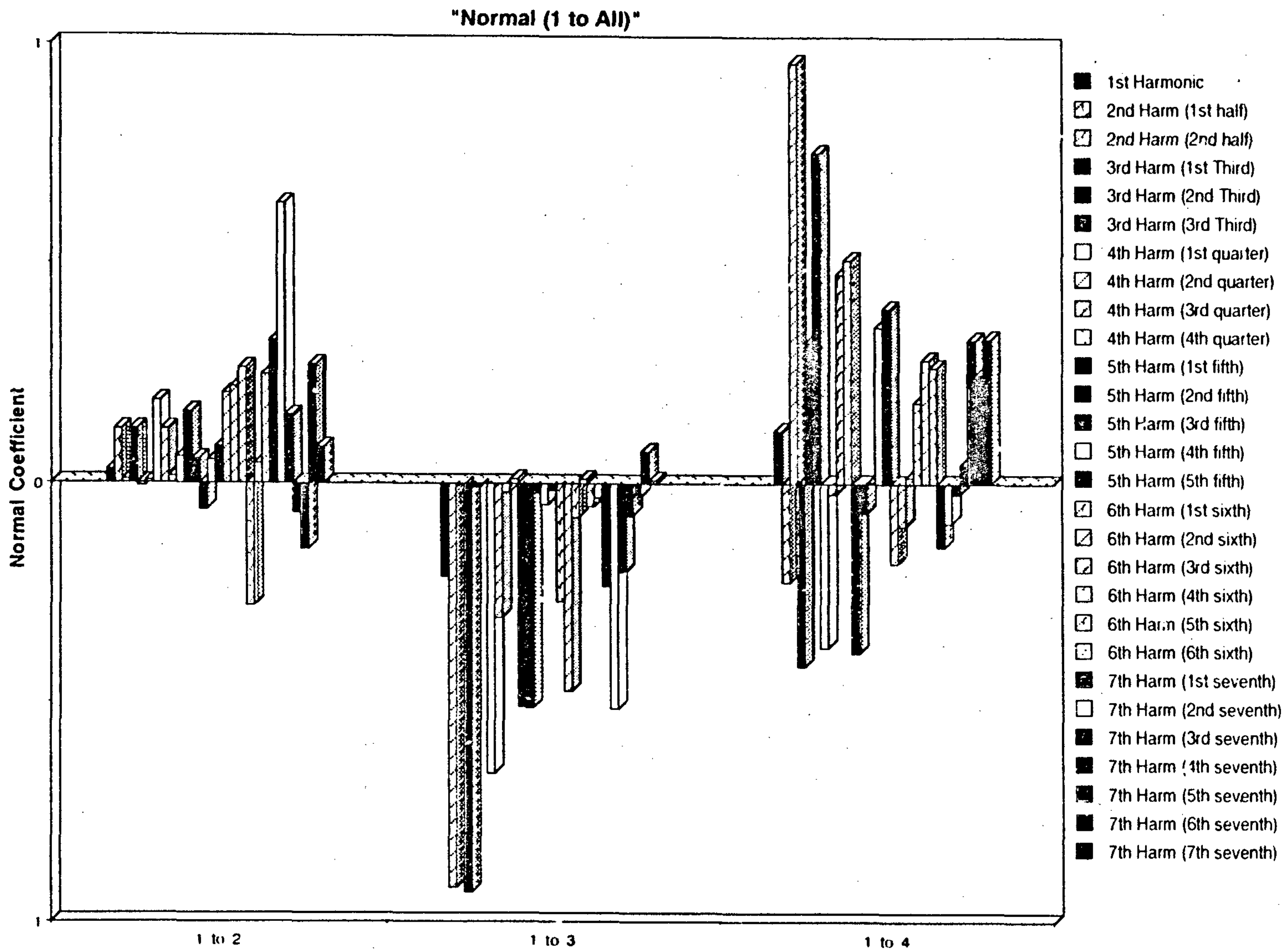
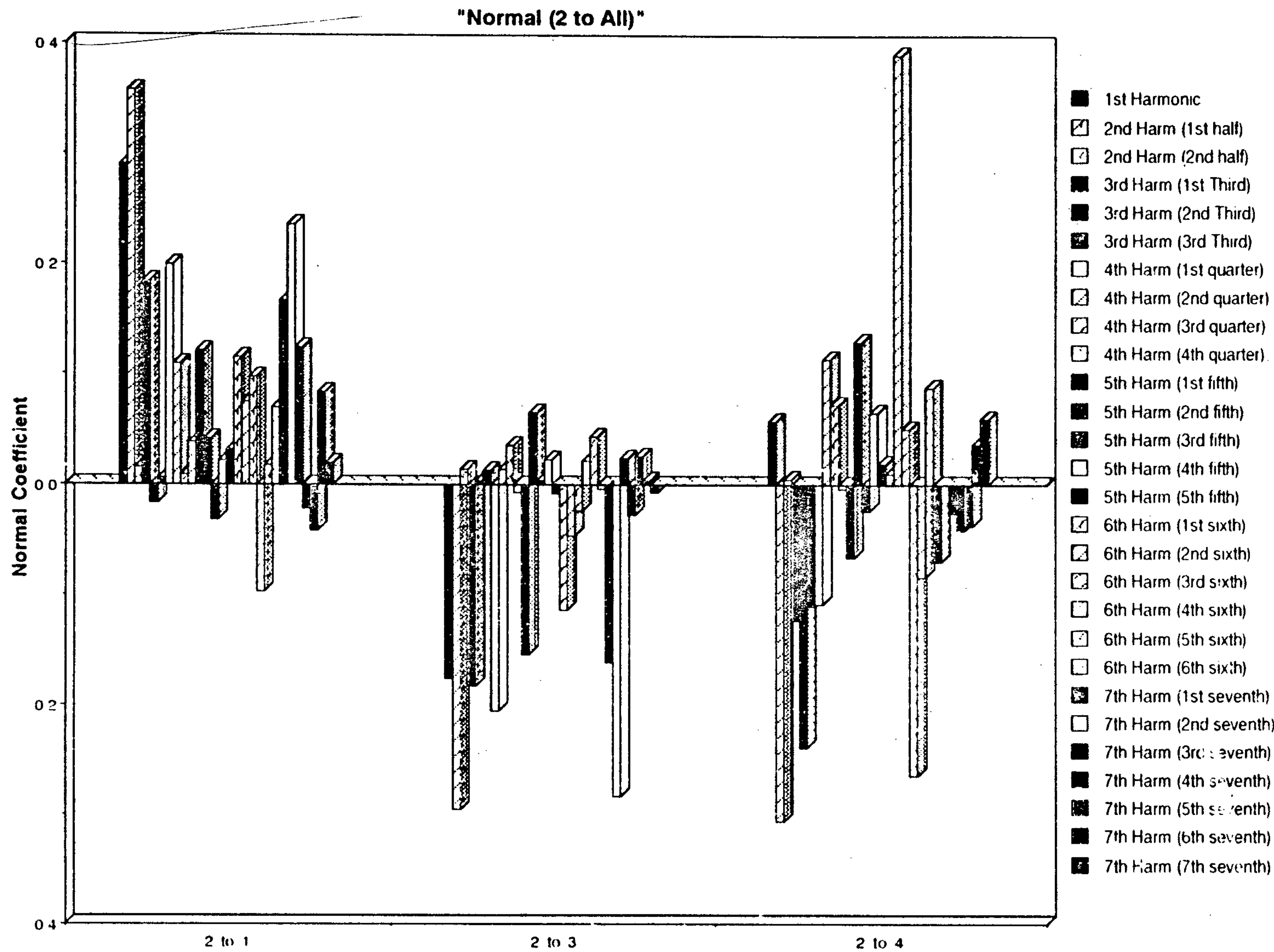

"Normal (1 to All)"
Normal Coefficient
1
0
1
1 to 2
1 to 3
1 to 4
1st Harmonic
2nd Harm (1st half)
2nd Harm (2nd half)
3rd Harm (1st Third)
3rd Harm (2nd Third)
3rd Harm (3rd Third)
4th Harm (1st quarter)
4th Harm (2nd quarter)
4th Harm (3rd quarter)
4th Harm (4th quarter)
5th Harm (1st fifth)
5th Harm (2nd fifth)
5th Harm (3rd fifth)
5th Harm (4th fifth)
5th Harm (5th fifth)
6th Harm (1st sixth)
6th Harm (2nd sixth)
6th Harm (3rd sixth)
6th Harm (4th sixth)
6th Harm (5th sixth)
6th Harm (6th sixth)
7th Harm (1st seventh)
7th Harm (2nd seventh)
7th Harm (3rd seventh)
7th Harm (4th seventh)
7th Harm (5th seventh)
7th Harm (6th seventh)
7th Harm (7th seventh)

Figure 7-11



"Normal (2 to All)"
Normal Coefficient
0.4
0.2
0.0
0.2
0.4
2 to 1
2 to 3
2 to 4
1st Harmonic
2nd Harm (1st half)
2nd Harm (2nd half)
3rd Harm (1st Third)
3rd Harm (2nd Third)
3rd Harm (3rd Third)
4th Harm (1st quarter)
4th Harm (2nd quarter)
4th Harm (3rd quarter)
4th Harm (4th quarter)
5th Harm (1st fifth)
5th Harm (2nd fifth)
5th Harm (3rd fifth)
5th Harm (4th fifth)
5th Harm (5th fifth)
6th Harm (1st sixth)
6th Harm (2nd sixth)
6th Harm (3rd sixth)
6th Harm (4th sixth)
6th Harm (5th sixth)
6th Harm (6th sixth)
7th Harm (1st seventh)
7th Harm (2nd seventh)
7th Harm (3rd seventh)
7th Harm (4th seventh)
7th Harm (5th seventh)
7th Harm (6th seventh)
7th Harm (7th seventh)

Figure 7-12

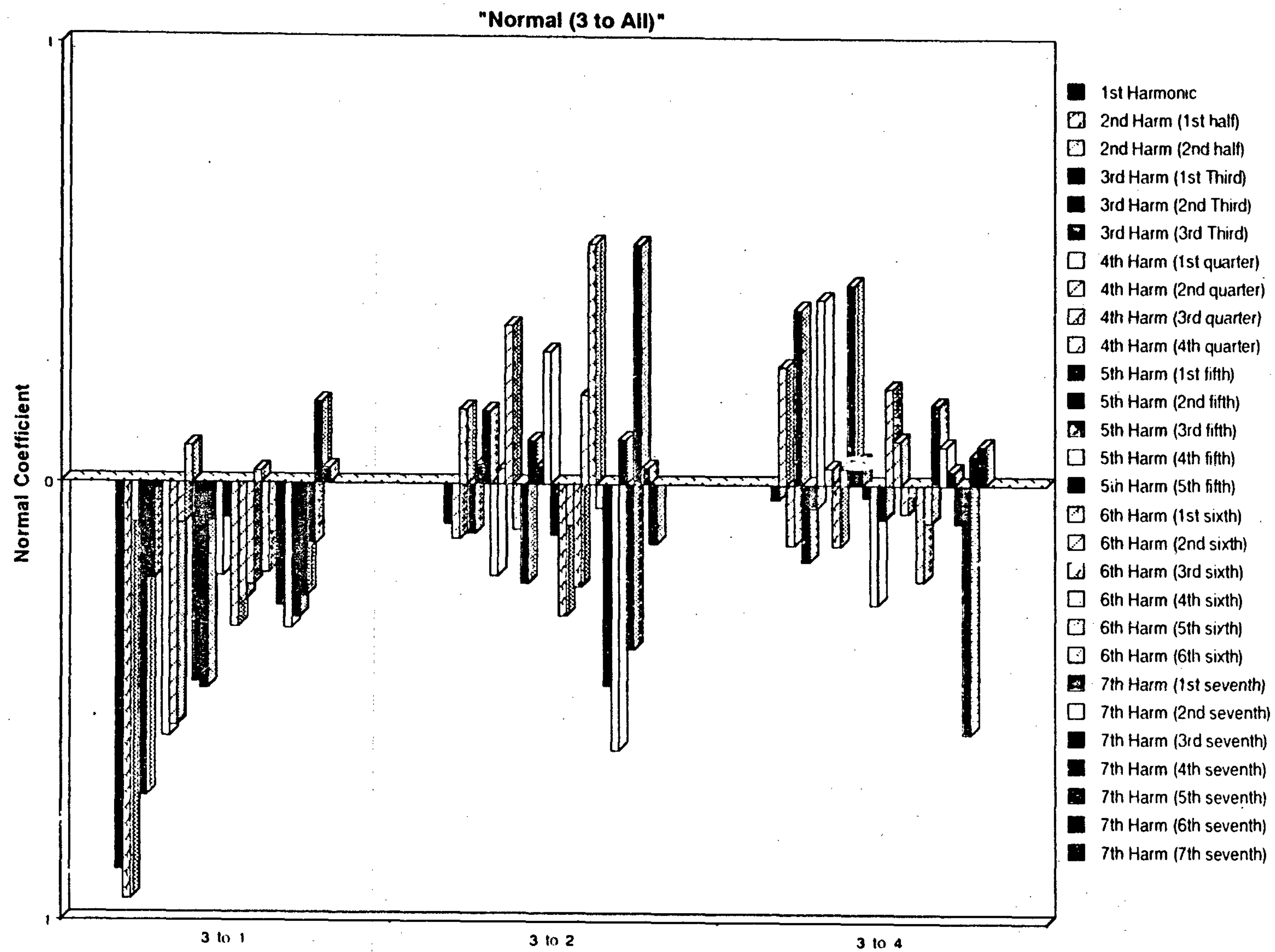

"Normal (3 to All)"
Normal Coefficient
1
0
1
3 to 1
3 to 2
3 to 4
1st Harmonic
2nd Harm (1st half)
2nd Harm (2nd half)
3rd Harm (1st Third)
3rd Harm (2nd Third)
3rd Harm (3rd Third)
4th Harm (1st quarter)
4th Harm (2nd quarter)
4th Harm (3rd quarter)
4th Harm (4th quarter)
5th Harm (1st fifth)
5th Harm (2nd fifth)
5th Harm (3rd fifth)
5th Harm (4th fifth)
5th Harm (5th fifth)
6th Harm (1st sixth)
6th Harm (2nd sixth)
6th Harm (3rd sixth)
6th Harm (4th sixth)
6th Harm (5th sixth)
6th Harm (6th sixth)
7th Harm (1st seventh)
7th Harm (2nd seventh)
7th Harm (3rd seventh)
7th Harm (4th seventh)
7th Harm (5th seventh)
7th Harm (6th seventh)
7th Harm (7th seventh)

Figure 7-13

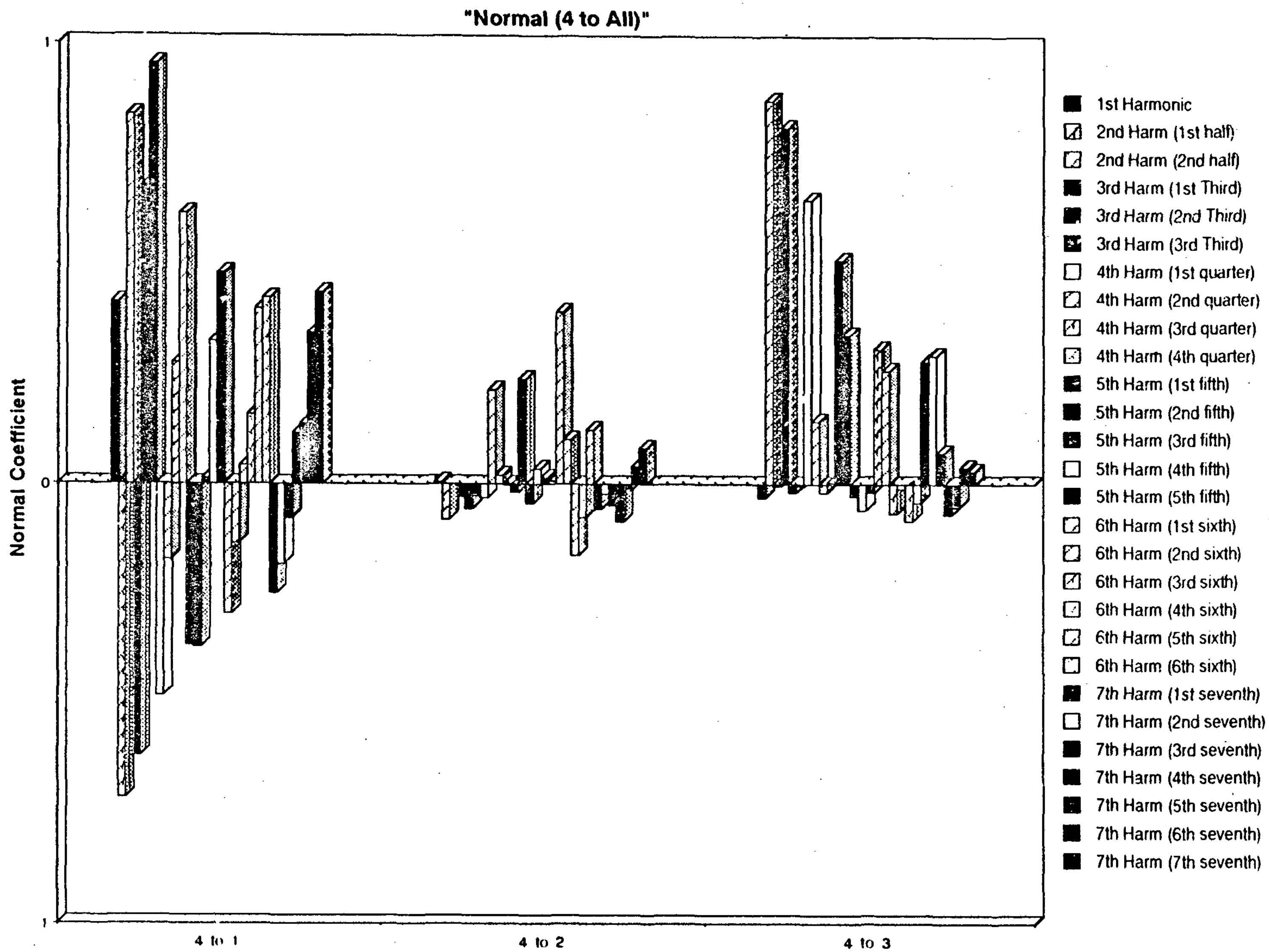

"Normal (4 to All)"
Normal Coefficient
1
0
1
4 to 1
4 to 2
4 to 3
1st Harmonic
2nd Harm (1st half)
2nd Harm (2nd half)
3rd Harm (1st Third)
3rd Harm (2nd Third)
3rd Harm (3rd Third)
4th Harm (1st quarter)
4th Harm (2nd quarter)
4th Harm (3rd quarter)
4th Harm (4th quarter)
5th Harm (1st fifth)
5th Harm (2nd fifth)
5th Harm (3rd fifth)
5th Harm (4th fifth)
5th Harm (5th fifth)
6th Harm (1st sixth)
6th Harm (2nd sixth)
6th Harm (3rd sixth)
6th Harm (4th sixth)
6th Harm (5th sixth)
6th Harm (6th sixth)
7th Harm (1st seventh)
7th Harm (2nd seventh)
7th Harm (3rd seventh)
7th Harm (4th seventh)
7th Harm (5th seventh)
7th Harm (6th seventh)
7th Harm (7th seventh)

Figure 7-14

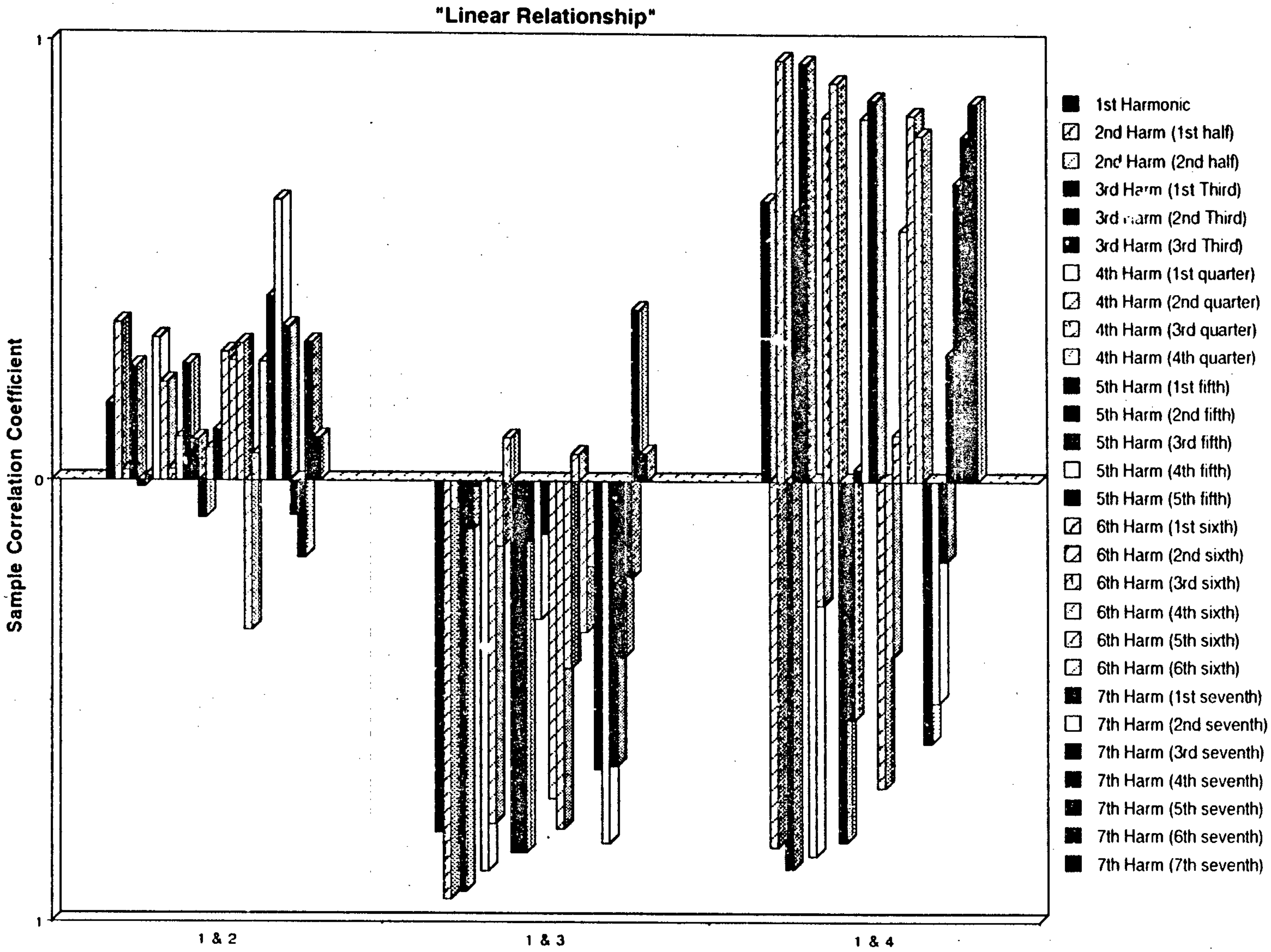

"Linear Relationship"
Sample Correlation Coefficient
1
0
1
1 & 2
1 & 3
1 & 4
1st Harmonic
2nd Harm (1st half)
2nd Harm (2nd half)
3rd Harm (1st Third)
3rd Harm (2nd Third)
3rd Harm (3rd Third)
4th Harm (1st quarter)
4th Harm (2nd quarter)
4th Harm (3rd quarter)
4th Harm (4th quarter)
5th Harm (1st fifth)
5th Harm (2nd fifth)
5th Harm (3rd fifth)
5th Harm (4th fifth)
5th Harm (5th fifth)
6th Harm (1st sixth)
6th Harm (2nd sixth)
6th Harm (3rd sixth)
6th Harm (4th sixth)
6th Harm (5th sixth)
6th Harm (6th sixth)
7th Harm (1st seventh)
7th Harm (2nd seventh)
7th Harm (3rd seventh)
7th Harm (4th seventh)
7th Harm (5th seventh)
7th Harm (6th seventh)
7th Harm (7th seventh)

Figure 7-15

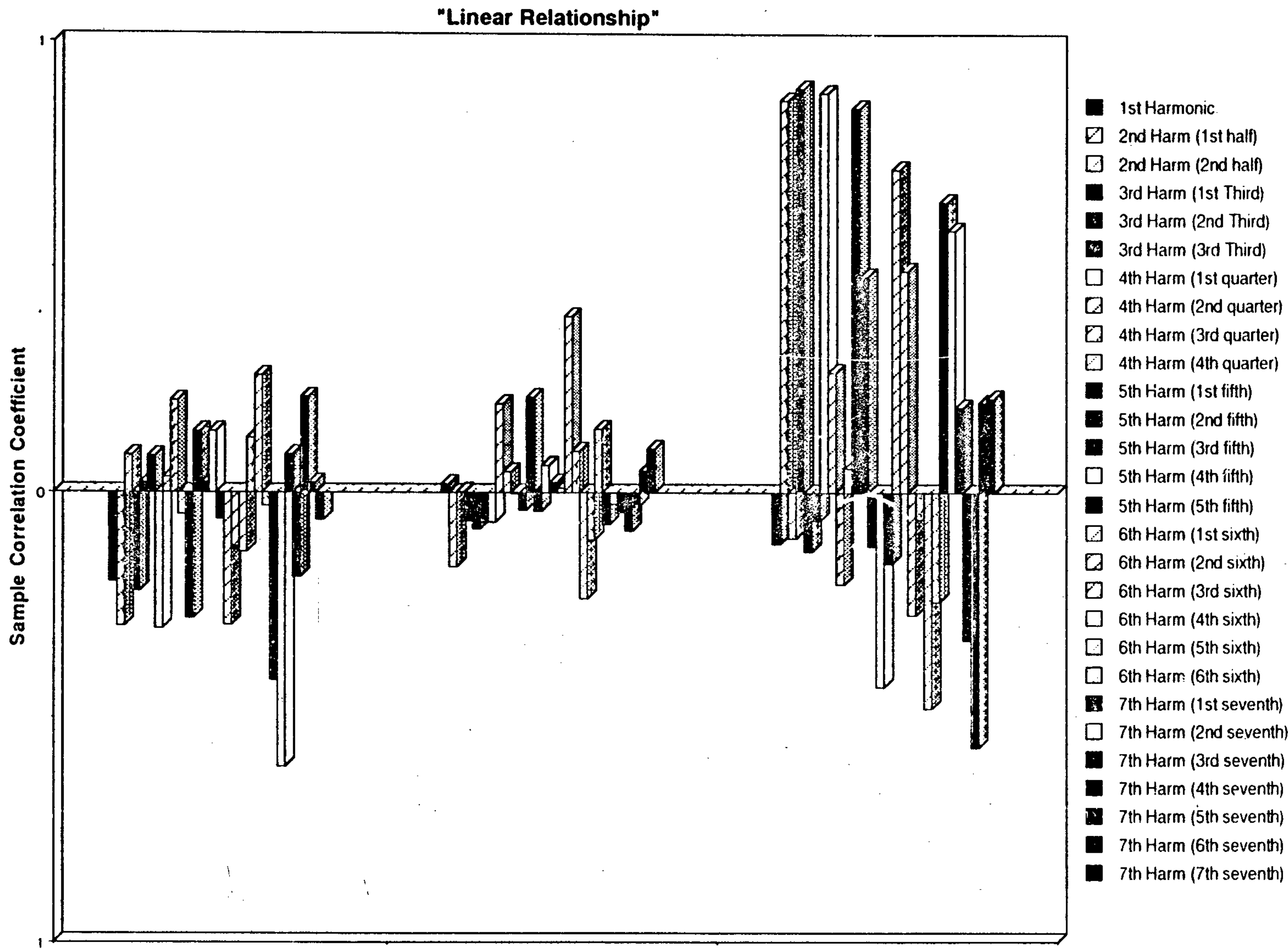

"Linear Relationship"
Sample Correlation Coefficient
1
0
1
2 & 3
2 & 4
3 & 4
1st Harmonic
2nd Harm (1st half)
2nd Harm (2nd half)
3rd Harm (1st Third)
3rd Harm (2nd Third)
3rd Harm (3rd Third)
4th Harm (1st quarter)
4th Harm (2nd quarter)
4th Harm (3rd quarter)
4th Harm (4th quarter)
5th Harm (1st fifth)
5th Harm (2nd fifth)
5th Harm (3rd fifth)
5th Harm (4th fifth)
5th Harm (5th fifth)
6th Harm (1st sixth)
6th Harm (2nd sixth)
6th Harm (3rd sixth)
6th Harm (4th sixth)
6th Harm (5th sixth)
6th Harm (6th sixth)
7th Harm (1st seventh)
7th Harm (2nd seventh)
7th Harm (3rd seventh)
7th Harm (4th seventh)
7th Harm (5th seventh)
7th Harm (6th seventh)
7th Harm (7th seventh)

Figure 7-16

STATISTICAL NEURAL NETWORK ARCHITECTURE

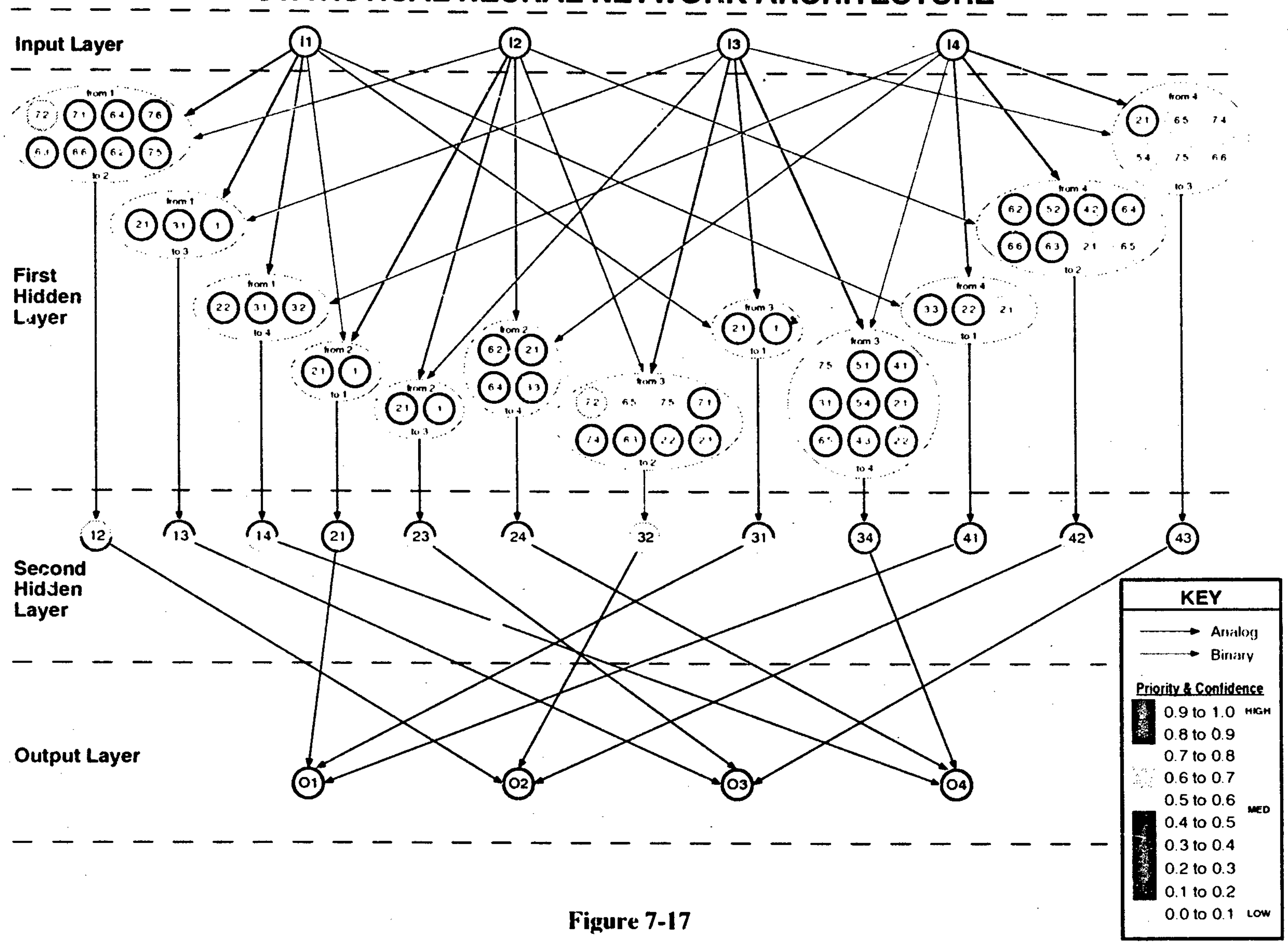


Figure 7-17

from the normal coefficients of Figures 7-11 through 7-14 and adding a node until all ranges are covered for each combination. For example, the oval on the far left in Figure 7-17, labeled "from 1" and "to 2", is constructed from the bar graph in Figure 7-11 labeled "1 to 2". The circles in each oval represent nodes. Each of these nodes has two inputs, shown coming into the top of each oval, and each node has one output, shown going into the second hidden layer. For clarity and convenience, not all of these connections have been shown. Each of the numbers in the nodes represent the range harmonic from which it came. For example, 7.2 found in the first node in Figure 7-17 came from Figure 7-11, graph 1 to 2, and signifies the seventh harmonic, second sector. It was chosen first because it has the greatest absolute value in Figure 7-11, graph 1 to 2. In this first oval, eight harmonics were needed to cover the complete range. If the first range harmonic had been chosen first, then there would have been no need to choose another, since the first harmonic covers the entire range.

After all of the nodes are created for each combination, the second hidden layer is built by having a node represent each combination from the first hidden layer. Each node in the second hidden layer has one input connection which comes directly from its combination in the first hidden layer. Each node here has one output which goes to the appropriate output node. The output layer contains the same number of nodes as the input layer, since the same parameters are being represented in these two layers. In this example, the four input and output nodes represent temperature, vibration, humidity, and number of failures, respectively. Each of the nodes in the output layer has three input connections which come from the combinations in the second hidden layer. This enables the network to predict values for up to three parameters not present at the input layer. For example, if you know the temperature, vibration, and humidity, but not the number of failures, then the network would provide its best guess for the number of failures.

7.2 Network Operation

Data enters the network at the input layer (see Figure 7-17). The minimum value within each range is subtracted from the data values at the input nodes. The resulting values are then multiplied by a weight which is equal to 1 over the range (i.e., 1/range). These weights are represented by the black arrow connections going from the input layer to the first hidden layer. This process normalizes the data so that the first hidden layer can handle it more readily. Each of the input nodes sends a binary signal to its corresponding outputs indicating whether it is on or off. A "1" is sent if on, and a "0" is sent if it is off. These binary signals are represented by the red arrow connections going from the input layer to the first hidden layer. The nodes in the first hidden layer are linear in nature with a dual threshold. Each node is activated only if the sum of two signals, one being the manipulated value and the other being the "0" or "1", falls between a lower and upper threshold. The upper threshold equals the sector divided by its harmonic (e.g., 7.2 has upper threshold equal to 2/7); the lower threshold equals the sector minus one, divided by its harmonic (e.g., 7.2 has lower threshold equal to 1/7). Therefore, only one node in a particular combination will fire if the harmonics do not overlap. Also, for each red connection that carries a "1", then all the nodes in that particular combination will be inactive because a "1" added to any input will always put the value above the upper threshold.

Activated nodes in the first hidden layer output their computed linear value along with the absolute value of their normal coefficients. Non-activated nodes output two zeros. The second hidden layer must then choose the computed linear value from each group which has the highest normal coefficient. The computed linear values chosen are converted back to their original states and passed through their respective linear regression equations to establish a best guess value for each remaining combination. The resulting numerical values are passed to the output layer, along with their normal coefficient and the absolute value of their correlation coefficient. At the output layer, the highest correlation coefficient indicates the network's solution to the problem. The network provides its answer with both the normal and correlation coefficients attached. The closer the normal and correlation coefficients are to one, the more likely the associated numerical answer is an accurate one. However, the coefficients do not represent a probability or a percentage of accuracy. The normal coefficient represents a priority rating, and the correlation coefficient represents a confidence rating. In Figure 7-17, the colored nodes in the first hidden layer represent the values of the priority ratings for each particular node, and the dual-colored nodes in the second hidden layer represent the values of the confidence ratings for that particular node, with the top color being the maximum expected and the bottom color the minimum expected. An example of the Statistical Neural Network is provided next to give more insight into the network's operation.

7.3 Statistical Neural Network in Action

This Statistical Neural Network will operate if one, two, or three inputs are provided. In this example, three inputs are given as: temperature = 380°K; vibration = 4 kHz; and humidity = 15%. The forth input parameter, number of failures, needs to be determined by the network. The first thing that happens when the inputs are passed through the input layer is to subtract the minimum value of the range from the original samples for all the inputs:

- **temperature (data1):** 380 - 299 = 81
- **vibration (data2):** 4 - 3.38 = 0.62
- **humidity (data3):** 15 - 5 = 10

Next, when these numbers are sent to the first hidden layer, they are multiplied by the values of the connection weights (here a connection weight is equivalent to 1 over the range):

- **data1:** 81 ÷ 102 = 0.79412
- **data2:** 0.61775 ÷ 3.32546 = 0.18576
- **data3:** 10 ÷ 47 = 0.21277

In the first hidden layer, these numbers are summed with the values carried on the red connections shown in Figure 7-17 (having values of "1" or "0"). In this example, since the first three input parameters are given, the only nodes in the first hidden layer which could possibly be active are those connected by the red arrows coming from input 4 (I4). The three active groups of nodes are the third, sixth and ninth groups (or ovals), labelled "from 1 to 4", "from 2 to 4", and "from 3 to 4", shown from left to right in Figure 7-17. The lower and upper thresholds for these nodes are listed below, first by group and then by priority:

- **"from 1 to 4" nodes:**
  - "2.2": lower = 0.5; upper = 1.0
  - "3.1": lower = 0.0; upper = 0.333
  - "3.2": lower = 0.333; upper = 0.667

- **"from 2 to 4" nodes:**
  - "6.2": lower = 0.167; upper = 0.333
  - "2.1": lower = 0.0; upper = 0.5
  - "6.4": lower = 0.5; upper = 0.667
  - "3.3": lower = 0.667; upper = 1.0

- **"from 3 to 4" nodes:**
  - "7.5": lower = 0.571; upper = 0.714
  - "5.1": lower = 0.0; upper = 0.2
  - "4.1": lower = 0.0; upper = 0.25
  - "3.1": lower = 0.0; upper = 0.333
  - "5.4": lower = 0.6; upper = 0.8
  - "2.1": lower = 0.0; upper = 0.5
  - "6.5": lower = 0.667; upper = 0.833
  - "4.3": lower = 0.5; upper = 0.75
  - "2.2": lower = 0.5; upper = 1.0

The only nodes from these groups that get activated are the nodes which contain the ranges that match the inputs:

- **"from 1 to 4" nodes:**
  - "2.2": lower = 0.5; upper = 1.0

- **"from 2 to 4" nodes:**
  - "6.2": lower = 0.167; upper = 0.333
  - "2.1": lower = 0.0; upper = 0.5

- **"from 3 to 4" nodes:**
  - "4.1": lower = 0.0; upper = 0.25
  - "3.1": lower = 0.0; upper = 0.333
  - "2.1": lower = 0.0; upper = 0.5

Next, the activated nodes pass the original fractional values along with the normal coefficients down to the second hidden layer:

- **"from 1 to 4" nodes:**
  - "2.2": fractional value: 0.79412
    normal coefficient: 0.960

- **"from 2 to 4" nodes:**
  - "6.2": fractional value: 0.18576
    normal coefficient: 0.390
  - "2.1": fractional value: 0.18576
    normal coefficient: 0.307

- **"from 3 to 4" nodes:**
  - "4.1": fractional value: 0.21277
    normal coefficient: 0.422
  - "3.1": fractional value: 0.21277
    normal coefficient: 0.400
  - "2.1": fractional value: 0.21277
    normal coefficient: 0.270

Within each group, only the fractional value with the highest normal coefficient will be used. At the second hidden layer, each remaining fractional value is transformed into its original state (0.79412 => 380; 0.18576 => 4; and 0.21277 => 15), and applied to its linear regression equation:

- **"14" node: "2.2":**
  linear regression line: $-71.2925 + 0.20598x$

- **"24" node: "6.2":**
  linear regression line: $-0.111996 + 0.319861x$

- **"34" node: "4.1":**
  linear regression line: $-3.11973 + 0.173626x$

The results of applying these parameters to their respective equations are listed below:

- **"14" node: "2.2":**
  $-71.2925 + 0.20598(380) = 6.9799$

- **"24" node: "6.2":**
  $-0.111996 + 0.319861(4) = 1.1674$

- **"34" node: "4.1":**
  $-3.11973 + 0.173626(15) = -0.5153$

These values, along with their associated normal and correlation coefficients, are sent to the output layer. Output node 4 (O4) receives the following data:

- **"14" node: "2.2":**
  answer: 6.9799
  normal coefficient: 0.960
  correlation coefficient: 0.960
- **"24" node: "6.2":**
  answer: 1.1674
  normal coefficient: 0.390
  correlation coefficient: 0.390
- **"34" node: "4.1":**
  answer: -0.5153
  normal coefficient: 0.422
  correlation coefficient: 0.889

At the output layer, the highest correlation coefficient indicates the answer. Thus, for this example, given the values of temperature = 380°K, vibration = 4 kHz, and humidity = 15%, the network determines the value for the number of failures to be 6.9799. This answer has associated with it normal and correlation coefficients which equal 0.96. The closer these coefficients are to 1.0, the more likely the associated numerical answer is an accurate one.

7.4 Summary

Statistics have been used in this application to help design a neural network whose architecture is tailored to input data. Statistical methods, if used properly, are a very powerful way to describe data. Neural networks, if used properly, can also be used to manipulate data and help draw some conclusions about it. Since statistics can be used to describe data, and since neural networks heavily involve statistics, it follows that neural network architectures and designs can better be accomplished by using statistical methods and techniques as tools during the design process. Data analysis can be performed more efficiently using network architectures and analysis techniques which have been tailored using appropriate statistics.

Simple statistical descriptors for finding the location, dispersion and correlation of data have been combined in the example presented here to form normal and correlation coefficients. These coefficients, along with range harmonics and linear regression lines, have been used to reformulate data into an integrated network called the Statistical Neural Network. The network provides the ability to predict unknown parameters given known parameters. The quality of the results depends upon the quality of the data used to form the network, as well as the kinds of data descriptors used in the network design. The Statistical Neural Network relies solely on statistical techniques at this time and does not yet incorporate important concepts such as probability and time. The most important contributions of this example are not necessarily the specific results presented, but to show how statistics can be used to develop neural network architectures based on data alone, and how appropriate data descriptors can be used to indicate and help determine natural tendencies in data.

8.0 Conclusion

This effort has addressed the feasibility of using neural network techniques in the development of automated Reliability/Maintainability/Testability (R/M/T) tools. The overall goal is to use neural network technology to perform R/M/T tasks in a quicker, easier, more accurate fashion. Work has included a feasibility study of neural network technology, investigation of links between neural networks and reliability, research on data aspects and related data analysis issues, and the development of a neural network whose architecture is based solely on statistics.

The feasibility portion of this effort focused on basic principles of neural networks and reliability. Attempts were aimed at realizing the potential benefits resulting from the combination of the two disciplines. Work involved gaining a comprehensive understanding of neural networks, with emphasis on fundamental concepts. Another contribution has been the realization that fundamental links exist between reliability and neural networks. These links are math-based, which implies that various data analysis and computational methods may be shared among the two disciplines. Common areas of mathematics have been identified, beginning with probability and statistics. The list expands to many other areas of math as well.

Research concerning the fundamental concepts of data has also been initiated here, with the hope of gaining insight and the understanding needed to automate certain functions required of intelligent information processing systems. While neural networks cannot provide the complete solution, it appears that they will be part of the solution, along with their more conventional counterparts. Fundamental issues concerning the nature of data can be modeled more naturally using neural-like rather than conventional techniques. More work is needed to explore the theories and concepts introduced here. The notion of automating such things as learning, communication, and decision-making is admittedly difficult and unusual, but it is not without hope. We hope that our contributions bring us closer to the goal.

We have also described the development of the Statistical Neural Network. This network relies on the statistical nature of data to build its architecture. Basic statistical descriptors are used to tune the network's architecture to input data, enabling more accurate analysis. This application indicates that statistics can be a powerful tool for describing natural tendencies in data. This in turn can lead to more efficient neural network designs and data analysis capabilities.

The results of this effort indicate that it would be very worthwhile to develop neural network techniques with the goal of improving the overall effectiveness of reliability analysis. With neural network technology gaining in popularity, the fundamental math-based similarities between neural networks and reliability imply that reliability science has much to gain in the long run. Groundwork laid in this effort will enable us to better understand and apply neural network techniques to R&M. The advantages of neural networks include the ability to learn or adapt, the ability to generalize, and parallel architecture. Potential benefits in the areas of data analysis and information processing include increased automated capabilities, improved analytical efficiency, increased accuracy and adaptability. Limitations of neural networks stem mainly from the fact that the technology's state of the art is relatively immature. Concepts are complex, abstract and dynamic, making network architectures and learning methods difficult to design and applications difficult to assess.

Recommendations for future work involve developing specific capabilities which exploit the advantages of neural network technology as applied to R/M/T problems. This involves developing automated tools and techniques for reliability analysis which handle data more naturally and efficiently. Reliability is not an exact science - it's data are subject to much interpretation. Neural network technology can provide techniques which are useful, effective, reliable and which currently do not exist.

9.0 *Final Remarks*

Neural networks are a computer technology which perform information processing in a manner unlike that of traditional computers. Neural networks are programmed differently, have a parallel rather than a serial nature, and are more tolerant to noisy or incomplete data. The potential of neural networks has become a hot topic lately, with many researchers working old or new problems using neural network techniques. However, given the flood of material on the subject, it is difficult to determine what the essential issues are, and what impact they may have of interest to the Air Force. Some people make wild claims about neural networks without sufficient evidence, while others make specific claims without addressing the big picture. Some researchers are doing excellent work. At this point in time, neural network technology is quite immature. Researchers are applying neural networks to many kinds of problems, but the problems are typically small or special purpose. Larger neural networks require extensive development, as do larger, more conventional computer methods. No single, useful, general purpose neural network exists today, at least in electronic form. Whatever form they exist in, neural networks should be applied appropriately.

One thing to note about neural networks is that certain fundamental laws or theories apply to all physical systems, neural networks included. Many of the basic concepts used in the design of neural networks already exist. Fundamental techniques are being combined in new ways to extend or improve existing data analysis methods. The concepts of control theory, adaptive systems, and statistical mechanics, to name a few, have provided some of the foundation for neural networks. Mathematics, physics, computer science and electronic engineering are relevant to neural networks as well as to reliability. Neural networks have been said to be a new name for old techniques. This may be true in part, but neural networks still have something to add to the technology base. The potential neural networks brings to reliability analysis, as well as to the entire computer industry, is quite large. The realization of this potential will occur in due time. The abstract nature of neural networks, as well as the complex mathematical and physical constraints which must be met for proper design, make the technology difficult to develop. Yet neural networks are here now and must be evaluated. At the very least, neural networks have thus far provided the means to combine the advantages of many technical disciplines under one roof.

Another important note is that much overlap exists in the fields of reliability and neural networks. Not enough interest has been paid on how neural networks may impact the field of reliability. Fundamental similarities exist in the math-based data analysis of neural networks and the mathematical models and methods used to perform reliability analysis. Reliability ultimately represents a measure of probability, and it involves using probabilistic and statistical methods to characterize reliability parameters using the best available data. Neural networks are a data analysis technique which inherently compile statistics and form probability distributions of incoming data patterns. A natural development would be to combine the disciplines of neural networks and reliability, resulting in automated R&M tools and techniques which are useful, practical and more powerful than existing ones.

The advantages of neural networks exist in theory, and even in practice, albeit on a small scale. We need to exploit the technology. But first we must develop it. The novelty of neural networks will diminish in time, but the underlying concepts will not go away. Concepts such as learning, generalizing, and parallel processing will continue to be of engineering interest. In the choice between performing a task manually or by computer, the automated method will win if it provides acceptable results. This is progress if the time and energy spent on the manual process is redirected toward tasks which man does better than machine (such tasks will always exist). Neural networks are a model used for processing and analyzing certain kinds of data. These models are approximations to reality, as all models are. The question here becomes: are neural networks a useful model? Can they improve on existing reliability analysis techniques? The answer is yes.

10.0 References

1. I. Aleksander, Editor, 1989, *Neural Computing Architectures: The Design of Brain-Like Machines*, MIT Press, Cambridge, MA.

2. W. Allman, 1989, *Apprentices of Wonder, Inside the Neural Network Revolution*, Bantam Books, NY.

3. D. Anderson, 1989, *Artificial Intelligence and Intelligent Systems: the Implications*, Ellis Horwood Limited, Chichester, England.

4. J. Anderson, 23 May 1991, *IEEE Videoconference*, "Neural Network Applications for the 1990's", Piscataway, NJ.

5. M. Arbib, 1964, *Brains, Machines, and Mathematics*, McGraw-Hill, NY.

6. M. Arbib, 1989, *The Metaphorical Brain 2. Neural Networks and Beyond*, John Wiley & Sons, Inc., NY.

7. N. Bourbaki, July 1990, *AI Expert*, "Turing, Searle, and Thought", Vol. 5, No. 7, pp. 52-59.

8. D. Burns, June 1991, Personal Communication.

9. J. Caroli, G. Lyne, J. Vaccaro, April 1990, *6th RAMCAD Technical Interchange Meeting*, "RADC Automated R/M/T Design Tools", Society of Logistic Engineers, Huntsville, AL, pp. 52-62.

10. M. Caudill and C. Butler, 1990, *Naturally Intelligent Systems*, MIT Press, Cambridge, MA.

11. D. Chait and M. Jensen, 1988, *MacBrain 2.0 User's Manual*, Neuronics, Inc., Cambridge, MA.

12. L. Doty, 1989, *Reliability for the Technologies*, Industrial Press Inc., NY.

13. A. Gleason, 1966, *Fundamentals of Abstract Analysis*, Addison-Wesley, MA.

14. J. Healy, December 1990, *IEEE Transactions on Reliability*, "Statistics, Reliability, and Uncertainty", Vol. 39, No. 5, p. 514.

15. D. Hebb, 1949, *The Organization of Behavior*, John Wiley & Sons, NY.

16. M. J. Kiemele and S. R. Schmidt, 1990, *Basic Statistics: Tools for Continuous Improvement*, Air Academy Press, Colorado Springs, CO.

17. C. Klimasauskas, J. Guiver and G. Pelton, 1989, *NeuralWorks Professional II: Neural Computing User's Manual*, NeuralWare, Inc., Pittsburg, PA, p. 25.

18. J. Klion, December 1989, Personal Communication.

19. R. Lippmann, April 1987, *IEEE ASSP Magazine*, "An Introduction to Computing with Neural Nets", pp. 4-22.

20. F. Machlup and U. Mansfield, Editors, 1983, *The Study of Information*, John Wiley & Sons, Inc., NY, p.28.

21. M. Minsky and S. Papert, 1969, *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, MA, pp. 18-20, 231-232.

22. G. Nagy, March 1991, *IEEE Transactions on Neural Networks*, "Neural Networks- Then and Now", Vol. 2, No. 2, pp. 316-318.

23. M. Roth, 1988, *Johns Hopkins APL Technical Digest*, "Neural-Network Technology and its Applications", Vol. 9, No. 3, pp. 242-253.

24. D. Rumelhart, J. McClelland, and the PDP Research Group, 1986, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, MIT Press, Cambridge, MA, p. 131.

25. T. Schwartz, 27 September 1989, Handout from: *IEEE Videoconference*, "Neural Networks: Capabilities & Applications, For Today and in the Future", p. 2. Tom J. Schwartz is the president of the Schwartz Associates, a consulting firm located in Mountain View, CA.

26. P. Simpson, 1990, *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations*, Pergamon Press, NY, pp. 30-135, 137.

27. A. Smith, 1984, *The Mind*, The Viking Press, NY, p.2.

28. H. Stack, J. Woods, 1986, *Probability, Random Processes, and Estimation Theory for Engineers*, Prentice-Hall, NJ.

29. B. W. Wah and C. V. Ramamoorthy, Editors, 1990, *Computers for Artificial Intelligence Processing*, John Wiley & Sons, Inc., NY.

30. N. Wiener, 1961, *Cybernetics or Control and Communication in the Animal and the Machine*, The MIT Press and John Wiley & Sons, Inc., NY, p. 114.

31. S. Wolfram, 1988, *Mathematica, A System for Doing Mathematics by Computer*, Addison-Wesley Publishing Company, Inc., Redwood City, CA.

32. R. Wood, Jr., 1986, *An Introduction To Reliability* (Draft), Air Force Institute of Technology, School of Systems & Logistics, Wright-Patterson AFB, OH, p.11.

33. L. Zadeh, 25 April 1991, Handout from: *IEEE Videoconference*, "Fuzzy Logic: Applications and Perspectives".

*U.S. GOVERNMENT PRINTING OFFICE: 1993-710-093-60161

MISSION

OF

ROME LABORATORY

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C^3I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.

END

FILMED

DATE: 4-93

DTIC